ISO TC184/SC4/WG3 /P19 N578

Date: 14 March 1997 **Supersedes SC4/ WG3 N 577 (P19)**

PRODUCT DATA REPRESENTATION AND EXCHANGE

Part: 214 **Title:** Application protocol: Core data for automotive mechanical design processes

Purpose of this document as it relates to the target document is:

X Primary Content
____ Issue Discussion
____ Alternate Proposal
____ Partial Content

Current Status: Committee draft

ABSTRACT:
This document defines an application protocol for Core Data for Automotive Mechanical Design Processes

KEYWORDS:
application protocol, automotive

Document Status/Dates (dd/mm/yy)

Part Documents		Other SC4 Documents
	Project Leader	
	Working Group Convenor	
	Qualification	
	Integration	
	Editing	

Owner/Editor: Juergen Mohrmann
Address: Daimler Benz
debis Systemhaus
Fasanenweg 9
D–70771 Leinfelden
Germany
Telephone/FAX: +49 711 972 -2149/-1905
E-mail: mohrmann@str.daimlerbenz.com

Alternate: Joerg Katzenmaier
Address: DiK
Technical University of Darmstadt
Petersenstr. 30
D–64287 Darmstadt
Germany
Telephone/FAX: +49 6151 16 -5145/-6854
E-mail: katmaier@dik.th-darmstadt.de

Comments to Reader

This document is the complete AP214 document prepared for distribution as the second CD release version.

Note: Finite element analysis has been put out of scope based on ballot issues received in the first CD ballot. The relevant issues #703, #1173, and #1251 are documented in the issue log ISO TC184/SC4/WG3 N575.

Contents

Page

1 Scope . . . 1

2 Normative references . . . 3

3 Definitions and abbreviations . . . 8
3.1 Terms defined in ISO 10303–1 . . . 8
3.2 Terms defined in ISO 10303–31 . . . 9
3.3 Terms defined in ISO 10303–42 . . . 9
3.4 Terms defined in ISO 10303–44 . . . 10
3.5 Terms defined in ISO 10303–45 . . . 11
3.6 Terms defined in ISO 10303–46 . . . 11
3.7 Terms defined in ISO 10303–47 . . . 11
3.8 Terms defined in ISO 10303–105 . . . 11
3.9 Terms defined in ISO 10303–202 . . . 12
3.10 Terms defined in ISO 13584–42 . . . 12
3.11 Other definitions . . . 12
3.11.1 body in white . . . 12
3.11.2 vehicle . . . 12
3.12 Abbreviations . . . 13

4 Information requirements . . . 14
4.1 Units of functionality . . . 15
4.1.1 surface_condition (C1) . . . 17
4.1.2 explicit_draughting (D1) . . . 17
4.1.3 associative_annotation (D2) . . . 18
4.1.4 external_reference_mechanism (E1) . . . 19
4.1.5 user_defined_feature (FF1) . . . 20
4.1.6 included_feature (FF2) . . . 20
4.1.7 generative_featured_shape (FF3) . . . 25
4.1.8 wireframe_model_2d (G1) . . . 25
4.1.9 wireframe_model_3d (G2) . . . 27
4.1.10 connected_surface_model (G3) . . . 28
4.1.11 faceted_b_rep_model (G4) . . . 30
4.1.12 b_rep_model (G5) . . . 31
4.1.13 compound_model (G6) . . . 34
4.1.14 csg_model (G7) . . . 36
4.1.15 geometrically_bounded_surface_model (G8) . . . 39
4.1.16 kinematics (K1) . . . 40
4.1.17 measured_data (MD1) . . . 42
4.1.18 item_property (PR1) . . . 43
4.1.19 geometric_presentation (P1) . . . 44
4.1.20 annotated_presentation (P2) . . . 45
4.1.21 shaded_presentation (P3) . . . 46
4.1.22 product_management_data (S1) . . . 46
4.1.23 element_structure (S2) . . . 47

4.1.24 item_definition_structure (S3) 48
4.1.25 effectivity (S4) 49
4.1.26 work_management (S5) 50
4.1.27 classification (S6) 51
4.1.28 specification_control (S7) 51
4.1.29 process_plan (S8) 54
4.1.30 dimension_tolerance (T1) 55
4.1.31 geometric_tolerance (T2) 56
4.2 Application objects 58
4.3 Application assertions 397

5 Application interpreted model 599
5.1 Mapping table 599
5.2 AIM EXPRESS short listing 1475
5.2.1 Fundamental concepts and assumptions 1486
5.2.2 automotive_design types 1497
5.2.3 automotive_design entities 1520
5.2.4 automotive_design rule definitions 1766
5.2.5 automotive_design function definitions 1935

6 Conformance requirements 1940
6.1 Conformance Class for component design with 3D shape representation (CC1) 1941
6.2 Conformance Class for assembly design with 3D shape representation (CC2) . 1943
6.3 Conformance Class for component drawings with wireframe or surface shape representation (CC3) 1943
6.4 Conformance Class for assembly drawings with wireframe, surface or solid shape representation (CC4) 1944
6.5 Conformance Class for styling data (CC5) 1945
6.6 Conformance Class for product data management without shape representation (CC6) 1945
6.7 Conformance Class for product data management with 3D shape representation (CC7) 1945
6.8 Conformance Class for configuration controlled design without shape representation (CC8) 1946
6.9 Conformance Class for configuration controlled design with 3D shape representation (CC9) 1946
6.10 Conformance Class for configuration controlled design with shape representation and draughting data (CC10) 1947
6.11 Conformance Class for process planning of components (CC11) 1948
6.12 Conformance Class for process planning of components with form feature and tolerance data (CC12) 1948
6.13 Conformance Class for effectivity controlled process planning of assemblies (CC13) 1949
6.14 Conformance Class for feature based design (CC14) 1950
6.15 Conformance Class for feature based design with flexible feature placement (CC15) 1951

6.16 Conformance Class for kinematic simulations for components and assemblies with 3d shape representation (CC16) . . . 1951
6.17 Conformance Class for measured data (CC17) . . . 1952
6.18 Conformance Class for data storage and retrieval systems (CC18) . . . 1952

Annexes

A AIM EXPRESS expanded listing . . . 1980
B AIM short names . . . 2305
C Implementation method-specific requirements . . . 2335
C.1 Requirements specific to the implementation method defined in ISO 10303–21 . 2335
D Protocol Implementation Conformance Statement (PICS) proforma . . . 2336
D.1 General conformance statements . . . 2337
D.1.1 Protocol implementation identification . . . 2337
D.1.2 Implementation method . . . 2337
D.1.3 Implemented conformance classes . . . 2337
D.2 Conformance statements for UoF C1 . . . 2338
D.3 Conformance statements for UoF D1 . . . 2338
D.4 Conformance statements for UoF D2 . . . 2338
D.5 Conformance statements for UoF E1 . . . 2338
D.6 Conformance statements for UoF FF1 . . . 2338
D.7 Conformance statements for UoF FF2 . . . 2339
D.8 Conformance statements for UoF G1 . . . 2339
D.9 Conformance statements for UoF G2 . . . 2339
D.10 Conformance statements for UoF G3 . . . 2340
D.11 Conformance statements for UoF G4 . . . 2340
D.12 Conformance statements for UoF G5 . . . 2340
D.13 Conformance statements for UoF G6 . . . 2341
D.14 Conformance statements for UoF G7 . . . 2341
D.15 Conformance statements for UoF G8 . . . 2342
D.16 Conformance statements for UoF K1 . . . 2342
D.17 Conformance statements for UoF MD1 . . . 2343
D.18 Conformance statements for UoF PR1 . . . 2343
D.19 Conformance statements for UoF P1 . . . 2344
D.20 Conformance statements for UoF P2 . . . 2344
D.21 Conformance statements for UoF P3 . . . 2344
D.22 Conformance statements for UoF S1 . . . 2344
D.23 Conformance statements for UoF S2 . . . 2345
D.24 Conformance statements for UoF S3 . . . 2345
D.25 Conformance statements for UoF S4 . . . 2345
D.26 Conformance statements for UoF S5 . . . 2345
D.27 Conformance statements for UoF S6 . . . 2346
D.28 Conformance statements for UoF S7 . . . 2346
D.29 Conformance statements for UoF S8 . . . 2347
D.30 Conformance statements for UoF T1 . . . 2347
D.31 Conformance statements for UoF T2 . . . 2347

E Information object registration . . . 2348
E.1 Document identification . . . 2348
E.2 Schema identification . . . 2348
F Application activity model . . . 2349
F.1 Application activity model definitions and abbreviations . . . 2349
F.2 Application activity model diagrams . . . 2355
G Application reference model . . . 2367
H AIM EXPRESS-G . . . 2419
J AIM EXPRESS listing . . . 2544
K ARM EXPRESS listing . . . 2546
L Bibliography . . . 2547

Index . . . 2548

Figures

1 Automotive mechanical design processes . . . xvi
2 Form features in panel defined by target boundaries and wall angles . . . 69
3 Angle_taper . . . 70
4 Angular_location_dimension . . . 72
5 Angular_location_dimension versus angular_size_dimension . . . 73
6 Annotation_subfigure for a hole on a technical drawing . . . 75
7 Predefined dimension symbols . . . 76
8 Predefined terminator symbols . . . 77
9 Bead_feature . . . 89
10 Chamfer . . . 96
11 Circular_boss . . . 98
12 Circular_closed_profile . . . 99
13 Circular_offset_pattern . . . 100
14 Circular_omit_pattern . . . 100
15 Circular_pattern with rotation of base feature . . . 103
16 Complete_circular_path . . . 116
17 Constant_radius_edge_round with offsets . . . 123
18 Constant_radius_fillet with offsets . . . 125
19 Predefined colours . . . 129
20 Predefined curve fonts . . . 130
21 Defined_thread . . . 143
22 Diameter_taper . . . 149
23 Components of a dimension on a drawing . . . 153
24 Draughting callouts on a drawing . . . 159
25 Edge transition . . . 164
26 Fill area styles . . . 180
27 Flat_slot_end_type . . . 182
28 General_boss . . . 186
29 General_closed_profile . . . 190
30 General_open_profile . . . 196
31 General_path . . . 197

32 General_pattern . . . 198
33 General_pocket_in_solid . . . 198
34 General_tolerances . . . 200
35 Geometrical_relationship . . . 209
36 Hardness . . . 213
37 Hole_bottom_condition . . . 215
38 Parameters defining a Joggle_feature . . . 231
39 Placement relationships kinematic pair frames and mating links . . . 235
40 Linear_path . . . 241
41 Ngon_profile . . . 255
42 Panel_hole_with_barring . . . 261
43 Partial_circular_path . . . 263
44 Partial_circular_profile . . . 264
45 Predefined point markers . . . 276
46 Position tolerancing with affected planes . . . 280
47 Example structure of Product_class objects in a hierarchy . . . 295
48 Radiused_slot_end_type . . . 312
49 Rectangular_closed_profile . . . 314
50 Rectangular_offset_pattern . . . 315
51 Rectangular_pattern . . . 318
52 Rib_feature . . . 324
53 Right_angular_wedge and its attributes . . . 325
54 Rounded_u_profile . . . 328
55 Same_time_machining_relationship . . . 330
56 Shape_aspect on a shaft . . . 334
57 Slot_feature . . . 341
58 Square_u_profile . . . 352
59 Indication of segment_size in a tolerance callout . . . 353
60 Presentation of Surface_conditions information on technical drawings . . . 360
61 Tee_profile . . . 367
62 Text appearance and rotation . . . 369
63 Tolerance_zone . . . 373
64 Vee_profile . . . 384
65 Woodruff_slot_end_type . . . 393
66 Interdependencies between UoFs . . . 1942
G.1 - G.50 ARM diagrams in EXPRESS . . . -
H.1 - H.124 AIM diagrams in EXPRESS . . . -

Tables

1 Logical grouping of UoFs . . . 15
2 Mapping table surface_condition UoF (C1) . . . 602
3 Mapping table explicit_draughting UoF (D1) . . . 621
4 Mapping table associative_annotation UoF (D2) . . . 636
5 Mapping table external_reference_mechanism UoF (E1) . . . 650
6 Mapping table user_defined_feature UoF (FF1) . . . 672

7 Mapping table included_feature UoF (FF2) 680
8 Mapping table generative_featured_shape UoF (FF3) 849
9 Mapping table wireframe_model_2d UoF (G1) 859
10 Mapping table wireframe_model_3d UoF (G2) 866
11 Mapping table connected_surface_model UoF (G3) 873
12 Mapping table faceted_b_rep_model UoF (G4) 890
13 Mapping table b_rep_model UoF (G5) 898
14 Mapping table compound_model UoF (G6) 915
15 Mapping table csg_model UoF (G7) 934
16 Mapping table geometrically_bounded_surface_model UoF (G8) 962
17 Mapping table kinematics UoF (K1) 970
18 Mapping table measured_data UoF (MD1) 982
19 Mapping table geometric_presentation UoF (P1) 984
20 Mapping table annotated_presentation UoF (P2) 1004
21 Mapping table shaded_presentation UoF (P3) 1015
22 Mapping table item_property UoF (PR1) 1021
23 Mapping table product_management_data UoF (S1) 1048
24 Mapping table element_structure UoF (S2) 1161
25 Mapping table item_definition_structure UoF (S3) 1184
26 Mapping table effectivity UoF (S4) 1209
27 Mapping table work_management UoF (S5) 1267
28 Mapping table classification UoF (S6) 1329
29 Mapping table specification_control UoF (S7) 1337
30 Mapping table process_plan UoF (S8) 1384
31 Mapping table dimension_tolerance UoF (T1) 1426
32 Mapping table geometric_tolerance UoF (T2) 1445
33 Usage of UoFs in Conformance Classes CC1-CC5 for CAD/CAM data exchange and sharing 1954
34 Usage of UoFs in Conformance Classes CC6-CC10 for product structure and configuration management data exchange and sharing 1955
35 Usage of UoFs in Conformance Classes CC11-CC15 for process planning and feature based design data exchange and sharing 1956
36 Usage of UoFs in Conformance Classes CC16-CC18 for simulation, quality control, and complete data storage and retrieval 1957
37 Usage of AIM elements in Conformance Classes 1958
B.1 AIM short names 2305
F.1 The AAM-data classes 2356
J.1 File names for schemas 2545

Foreword

The International Organization for Standardization (ISO) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

Draft International Standards adopted by technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75% of the member bodies casting a vote.

International Standard ISO 10303–214 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration,* Subcommittee SC4, *Industrial data.*

ISO 10303 consists of the following parts under the general title *Industrial automation systems and integration – Product data representation and exchange:*

- Part 1, Overview and fundamental principles;
- Part 11, Description methods: The EXPRESS language reference manual;
- Part 12, Description method: The EXPRESS-I language reference manual;
- Part 21, Implementation methods: Clear text encoding of the exchange structure;
- Part 22, Implementation method: Standard data access interface specification;
- Part 23, Implementation method: C++ language binding to the standard data access interface;
- Part 24, Implementation method: C language binding to the standard data access interface;
- Part 26, Implementation method: Interface definition language binding to the standard data access interface;
- Part 31, Conformance testing methodology and framework: General concepts;
- Part 32, Conformance testing methodology and framework: Requirements on testing laboratories and clients;
- Part 33, Conformance testing methodology and framework: Structure and use of abstract test suites;
- Part 34, Conformance testing methodology and framework: Abstract test methods;

– Part 35, Conformance testing methodology and framework: Abstract test methods for SDAI implementations;

– Part 41, Integrated generic resources: Fundamentals of product description and support;

– Part 42, Integrated generic resources: Geometric and topological representation;

– Part 43, Integrated generic resources: Representation structures;

– Part 44, Integrated generic resources: Product structure configuration;

– Part 45, Integrated generic resource: Materials;

– Part 46, Integrated generic resources: Visual presentation;

– Part 47, Integrated generic resource: Shape variation tolerances;

– Part 49, Integrated generic resource: Process structure and properties;

– Part 101, Integrated application resource: Draughting;

– Part 104, Integrated application resource: Finite element analysis;

– Part 105, Integrated application resource: Kinematics;

– Part 106, Integrated application resource: Building construction core model;

– Part 201, Application protocol: Explicit draughting;

– Part 202, Application protocol: Associative draughting;

– Part 203, Application protocol: Configuration controlled design;

– Part 204, Application protocol: Mechanical design using boundary representation;

– Part 205, Application protocol: Mechanical design using surface representation;

– Part 207, Application protocol: Sheet metal die planning and design;

– Part 208, Application protocol: Life cycle management - Change process;

– Part 209, Application protocol: Composite and metallic structural analysis and related design;

– Part 210, Application protocol: Electronic assembly, interconnect, and packaging design;

– Part 212, Application protocol: Electrotechnical design and installation;

- Part 213, Application protocol: Numerical control process plans for machined parts;
- Part 214, Application protocol: Core data for automotive mechanical design processes;
- Part 215, Application protocol: Ship arrangement;
- Part 216, Application protocol: Ship moulded forms;
- Part 217, Application protocol: Ship piping;
- Part 218, Application protocol: Ship structures;
- Part 220, Application protocol: Process planning, manufacture, and assembly of layered electronic products;
- Part 221, Application protocol: Functional data and their schematic representation for process plant;
- Part 222, Application protocol: Exchange of product data for composite structures;
- Part 223, Application protocol: Exchange of design and manufacturing product information for casting parts;
- Part 224, Application protocol: Mechanical product definition for process plans using machining features;
- Part 225, Application protocol: Building elements using explicit shape representation;
- Part 226, Application protocol: Ship mechanical systems;
- Part 227, Application protocol: Plant spatial configuration;
- Part 228, Application protocol: Building services: Heating, ventilation, and air conditioning;
- Part 229, Application protocol: Exchange of design and manufacturing product information for forged parts;
- Part 230, Application protocol: Building structural frame: Steelwork;
- Part 231, Application protocol: Process engineering data: Process design and process specification of major equipment;
- Part 232, Application Protocol: Technical data packaging core information and exchange;
- Part 301, Abstract test suite: Explicit draughting;

– Part 302, Abstract test suite: Associative draughting;

– Part 303, Abstract test suite: Configuration controlled design;

– Part 304, Abstract test suite: Mechanical design using boundary representation;

– Part 305, Abstract test suite: Mechanical design using surface representation;

– Part 307, Abstract test suite: Sheet metal die planning and design;

– Part 308, Abstract test suite: Life cycle management - Change process;

– Part 309, Abstract test suite: Composite and metallic structural analysis and related design;

– Part 310, Abstract test suite: Electronic assembly, interconnect, and packaging design;

– Part 312, Abstract test suite: Electrotechnical design and installation;

– Part 313, Abstract test suite: Numerical control process plans for machined parts;

– Part 314, Abstract test suite: Core data for automotive mechanical design;

– Part 315, Abstract test suite: Ship arrangement;

– Part 316, Abstract test suite: Ship moulded forms;

– Part 317, Abstract test suite: Ship piping;

– Part 318, Abstract test suite: Ship structures;

– Part 320, Abstract test suite: Process planning, manufacture, and assembly of layered electronic products;

– Part 321, Abstract test suite: Functional data and their schematic representation for process plant;

– Part 322, Abstract test suite: Exchange of product data for composite structures;

– Part 323, Abstract test suite: Exchange of design and manufacturing product information for casting parts;

– Part 324, Abstract test suite: Mechanical product definition for process plans using machining features;

– Part 325, Abstract test suite: Building elements using explicit shape representation;

- Part 326, Abstract test suite: Ship mechanical systems;

- Part 327, Abstract test suite: Plant spatial configuration;

- Part 328, Abstract test suite: Building services: Heating, ventilation, and air conditioning;

- Part 329, Abstract test suite: Exchange of design and manufacturing product information for forged parts;

- Part 330, Abstract test suite: Building structural frame: Steelwork;

- Part 331, Abstract test suite: Process engineering data: Process design and process specification of major equipment;

- Part 332, Abstract test suite: Technical data packaging core information and exchange;

- Part 501, Application interpreted construct: Edge-based wireframe;

- Part 502, Application interpreted construct: Shell-based wireframe;

- Part 503, Application interpreted construct: Geometrically bounded 2D wireframe;

- Part 504, Application interpreted construct: Draughting annotation;

- Part 505, Application interpreted construct: Drawing structure and administration;

- Part 506, Application interpreted construct: Draughting elements;

- Part 507, Application interpreted construct: Geometrically bounded surface;

- Part 508, Application interpreted construct: Non-manifold surface;

- Part 509, Application interpreted construct: Manifold surface;

- Part 510, Application interpreted construct: Geometrically bounded wireframe;

- Part 511, Application interpreted construct: Topologically bounded surface;

- Part 512, Application interpreted construct: Faceted boundary representation;

- Part 513, Application interpreted construct: Elementary boundary representation;

- Part 514, Application interpreted construct: Advanced boundary representation;

- Part 515, Application interpreted construct: Constructive solid geometry;

- Part 517, Application interpreted construct: Mechanical design geometric presentation;
- Part 518, Application interpreted construct: Mechanical design shaded representation.

The structure of this International Standard is described in ISO 10303-1. The numbering of the parts of this International Standard reflects its structure:

- Parts 11 to 13 specify the description methods,
- Parts 21 to 26 specify the implementation methods,
- Parts 31 to 35 specify the conformance testing methodology and framework,
- Parts 41 to 49 specify the integrated generic resources,
- Parts 101 to 106 specify the integrated application resources,
- Parts 201 to 232 specify the application protocols,
- Parts 301 to 332 specify the abstract test suites, and
- Parts 501 to 518 specify the application interpreted constructs.

Should further parts of ISO 10303 be published, they will follow the same numbering pattern.

Annexes A, B, C, D, and E are an integral part of this part of ISO 10303. Annexes F, G, H, J, K, and L are for information only.

Introduction

ISO 10303 is an International Standard for the computer-interpretable representation and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

This International Standard is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application interpreted constructs, application protocols, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303–1. This part of ISO 10303 is a member of the application protocol series.

This part of ISO 10303 specifies an application protocol (AP) for automotive mechanical design processes. This application protocol defines the context, scope, and information requirements for various development stages during the design of a vehicle and specifies the integrated resources necessary to satisfy these requirements. This application protocol addresses the requirements of the automotive industry covering cars, trucks, busses, and motorcycles.

NOTE 1 – This application protocol may also be applicable to other forms of vehicles, such as rail, agricultural, and construction vehicles.

A typical vehicle consists of the body, power train, chassis, and interior parts.

The body describes the body shell and the body in white.

EXAMPLE 1 – Typical constituents of the body are frame, doors, roof, engine hood, windshield, or bumpers.

The power train includes the engine, transmission, and drive line.

EXAMPLE 2 – Typical constituents of the power train are gear box, pistons, and exhaust pipe.

The chassis includes all parts which are responsible for the vehicles handling.

EXAMPLE 3 – Typical constituents of the chassis are the front axle, rear axle, steering, suspension, wheels, shock absorbers, and brakes.

Interior parts include parts that are built into the interior of the vehicle and not related to the power train.

EXAMPLE 4 – Typical interior parts are seats, instrument panel, door panels, the mechanism for power windows, air conditioning system, or stereo equipment.

Only the mechanical aspects of a vehicle and its components are covered by this AP; the functional aspects, such as electronic, hydraulic, and pneumatic, are not covered.

EXAMPLE 5 – For a battery or a switch, product characteristics such as shape, material, or mass are covered but the electrical functionality is not covered.

Finite element analysis (FEA) is not supported by this AP.

NOTE 2 – Finite element analysis has been put out ouf scope based on ballot issues received in the first CD ballot. The decision to take FEA out of scope was mainly based on the fact that only two nations had an active interest in this topic.

Figure 1 shows the processes included in the development of a vehicle. For further explanation of the processes some examples of significant activities or data are given. The illustrations are

taken from car body development. The application activity model given in annex F provides a further detailed graphical representation of the processes and information flows.

These processes include:

- product definition;
- styling;
- design;
- evaluation;
- production planning;
- tool design;
- tool manufacturing;
- quality control.

Application protocols provide the basis for developing implementations of ISO 10303 and abstract test suites for the conformance testing of AP implementations.

Clause 1 defines the scope of the application protocol and summarizes the functionality and data covered by the AP. Clause 3 lists the words defined in this part of ISO 10303 and gives pointers to words defined elsewhere. An application activity model that is the basis for the definition of the scope is provided in annex F. The information requirements of the application are specified in clause 4 using terminology appropriate to the application. A graphical representation of the information requirements, referred to as the application reference model, is given in annex G.

Resource constructs are interpreted to meet the information requirements. This interpretation produces the application interpreted model (AIM). This interpretation, given in 5.1, shows the correspondence between the information requirements and the AIM[1)]. The short listing of the AIM specifies the interface to the integrated resources and is given in 5.2. Note that the definitions and EXPRESS provided in the the integrated resources for constructs used in the AIM may include select list items and subtypes which are not imported into the AIM. The expanded listing given in annex A contains the complete EXPRESS for the AIM without annotation. A graphical representation of the AIM is given in annex H. Additional requirements for specific implementation methods are given in annex D.

[1)]For the interpretation process the New Editions of parts 41, 43, and 44 and the new AICs for associative draughting elements and for geometric tolerances as balloted as New Work Items in spring 1997 are used.

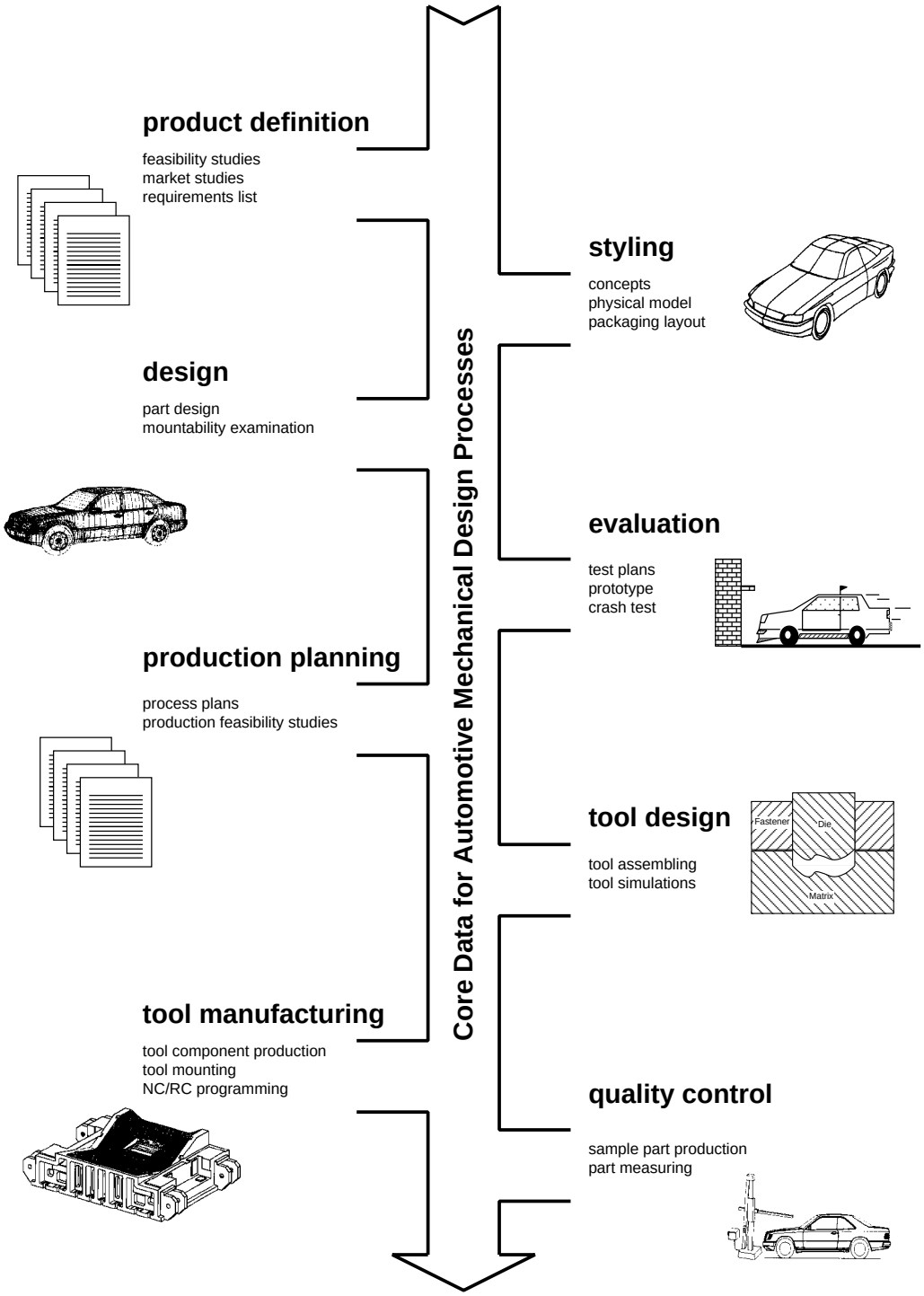


Figure 1 – Automotive mechanical design processes

Industrial automation systems and integration — Product data representation and exchange — Part 214: Application protocol: Core data for automotive mechanical design processes

1 Scope

This part of ISO 10303 specifies the use of the integrated resources necessary for the scope and information requirements for the exchange of information between various applications which support the development process of the mechanical aspects of a vehicle.

NOTE – The application activity model in annex F provides a graphical representation of the processes and information flows which are the basis for the definition of the scope of this part of ISO 10303.

The following are within the scope of this part of ISO 10303:

a) products of automotive manufacturers and suppliers that include parts, assemblies of parts, tools, and assemblies of tools. The parts include the constituents of the car body, power train, chassis, and interior. The tools include those specific to the product produced and used by various manufacturing technologies, such as shaping, transforming, separating, coating, or fitting;

EXAMPLE 6 – Typical technologies for primary shaping are molding or casting, for transforming are bending or stamping, for separating are milling or lathing, for coating are painting or surface coating, and for fitting are welding or riveting.

b) process plan information to manage the relationships between parts and the tools used to manufacture them and to manage the relationships between intermediate stages of parts or tools, referred to as in-process parts;

c) product definition data and configuration control data pertaining to the design phase of a product's development;

d) changes of a design, including tracking of the versions of a product and data related to the documentation of the change process;

e) identification of alternate representations of parts and tools during the design phase;

f) identification of standard parts, based on international, national, or industrial standards, and of library parts, based on company or project conventions;

g) release and approval data for various kinds of product data;

h) data that identify the supplier of a product and any related contract information;

i) any of eight types of representation of the shape of a part or tool:

1) 2D–wireframe representation;

2) 3D–wireframe representation;

3) geometrically bounded surface representation;

4) topologically bounded surface representation;

5) faceted–boundary representation;

6) boundary representation;

7) compound shape representation;

8) constructive solid geometry representation.

j) shape representation of parts or tools that is a combination of any two or more of these eight types of shape representation;

k) data that pertains to the presentation of the shape of the product;

l) representation of portions of the shape of a part or a tool by form features;

m) product documentation represented by explicit and associative draughting;

n) references to product documentation represented in a form or format other than that specified by ISO 10303;

EXAMPLE 7 – Other forms or formats include physical clay models, digital data in other standard formats such as NC–data according to ISO 6983 [6], or text data according to ISO/IEC 8879 (SGML) [8].

o) simulation data for the description of kinematic structures and configurations of discrete tasks;

EXAMPLE 8 – The simulation data for a windshield wiper includes the geometry of the windshield as well as the kinematic structure of the wiper including all necessary links and joints.

p) properties of parts or tools;

q) surface conditions;

r) dimensional and geomterical tolerance data.

The following are outside the scope of this part of ISO 10303:

a) product definition data pertaining to any life cycle phase of a product not related to design;

b) business or financial data for the management of a design project;

c) a general parametric representation of the shape of the part or tool;

d) data describing the pneumatic, hydraulic, electric, or electronic functions of a product;

e) continuous kinematics simulations over time;

f) data describing the input or output of finite element analysis.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO 10303. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO 10303 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 286–1:1988, *ISO system of limits and fits – Part 1: Bases of tolerances, deviations, and fits.*

ISO 286–2:1988, *ISO system of limits and fits – Part 2: Tables of standard tolerance grades and limit deviations for holes and shafts.*

ISO/DIS 639–2:—[2], *Codes for the representation of names of languages – Part 2: Alpha-3 code.*

ISO 965–1:1980, *ISO general purpose metric screw threads – Tolerances – Part 1: Principles and basic data.*

ISO 1101:1983, *Technical drawings – Geometric tolerancing – Tolerancing of form, orientation, location and run-out – Generalities, definitions, symbols, indications on drawings.*

ISO 2692:1988, *Technical drawings – Geometric tolerancing – Maximum material principle.*

ISO 2768–1:1989, *General tolerances – Part 1: Tolerances for linear and angular dimensions without individual tolerance indications.*

[2] To be published.

ISO 2768–2:1989, *General tolerances – Part 2: Geometrical tolerances for features without individual tolerance indications.*

ISO 3098–1:1974, *Technical drawings; lettering; part 1: Currently used characters.*

ISO 3098–2:1984, *Technical drawings; lettering; part 2: Greek characters.*

ISO 5458:1987, *Technical drawings – Geometric tolerancing – Positional tolerancing.*

ISO 5459:1981, *Technical drawings – Geometric tolerancing – Datums and datum-systms for geometrical tolerances.*

ISO 8015:1985, *Technical drawings – Fundamental tolerancing principle.*

ISO 8824–1:1994, *Information Technology – Open systems interconnection – Abstract syntax notation one (ASN.1) – Part 1: Specification of basic notation.*

ISO 10303–1:1994, *Industrial automation systems and integration*
– Product data representation and exchange:
– Overview and fundamental principles

ISO 10303–11:1994, *Industrial automation systems and integration*
– Product data representation and exchange:
– Description methods: The EXPRESS *language reference manual*

ISO 10303–21:1994, *Industrial automation systems and integration*
– Product data representation and exchange:
– Implementation methods: Clear text encoding of the exchange structure

ISO 10303–22:—[2)], *Industrial automation systems and integration*
– Product data representation and exchange:
– Implementation methods: Standard data access interface

ISO 10303–31:1994, *Industrial automation systems and integration*
– Product data representation and exchange:
– Conformance testing methodology and framework: General concepts

ISO 10303–41:—[2)], *Industrial automation systems and integration*
– Product data representation and exchange:
– Integrated generic resource: Fundamentals of product description and support
– ISO TC 184/SC4 N539

ISO 10303–42:1994, *Industrial automation systems and integration*
– Product data representation and exchange:
– Integrated generic resources: Geometric and topological representation

ISO 10303–43:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Integrated generic resource: Representation structures
– ISO TC 184/SC4 N540

ISO 10303–44:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Integrated generic resource: Product structure configuration
– ISO TC 184/SC4 N541

ISO 10303–45:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Integrated generic resources: Materials

ISO 10303–46:1994, *Industrial automation systems and integration*
– Product data representation and exchange:
– Integrated generic resources: Visual presentation

ISO 10303–47:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Integrated generic resources: Shape variation tolerances

ISO 10303–49:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Integrated generic resources: Process structure and properties

ISO 10303–101:1994, *Industrial automation systems and integration*
– Product data representation and exchange:
– Integrated application resources: Draughting

ISO 10303–105:1996, *Industrial automation systems and integration*
– Product data representation and exchange:
– Integrated application resources: Kinematics

ISO 10303–201:1994, *Industrial automation systems and integration*
– Product data representation and exchange:
– Application protocol: Explicit draughting

ISO 10303–202:1996, *Industrial automation systems and integration*
– Product data representation and exchange:
– Application protocol: Associative draughting

ISO 10303–501:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Edge-based wireframe

ISO 10303–503:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Geometrically bounded 2d wireframe

ISO 10303–504:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Draughting annotation

ISO 10303–506:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Draughting elements

ISO 10303–507:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Geometrically bounded wireframe

ISO 10303–508:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Non-manifold surface

ISO 10303–509:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Manifold surface

ISO 10303–510:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Geometrically bounded wireframe

ISO 10303–511:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Topologically bounded surface

ISO 10303–512:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Faceted boundary representation

ISO 10303–514:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Advanced boundary representation

ISO 10303–515:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Constructive solid geometry

ISO 10303–517:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Mechanical design geometric presentation

ISO 10303–5x1:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Geometric tolerances
– ISO TC 184/SC4 N542

ISO 10303–5x2:—[2], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Associative draughting elements
– ISO TC 184/SC4 N543

ISO 10578:1992, *Technical drawings – Tolerancing of orientation and location – Projected tolerance zone.*

ISO 13584–42:—[2], *Industrial automation systems and integration*
– Parts library:
– Part 42: Description methodology: Methodology for structuring part families

3 Definitions and abbreviations

For the purpose of this part of ISO 10303, the following definitions and abbreviations apply.

3.1 Terms defined in ISO 10303–1

This part of ISO 10303 makes use of the following terms defined in ISO 10303–1:

- application;
- application activity model (AAM);
- application interpreted model (AIM);
- application object;
- application protocol (AP);
- application reference model (ARM);
- assembly;
- component;
- conformance class;
- conformance requirements;
- data;
- data exchange;
- generic resource;
- implementation method ;
- information;
- information model;
- integrated resource;
- PICS proforma;
- presentation;

- product;
- product data;
- product information;
- resource construct;
- structure;
- unit of functionality (UoF).

3.2 Terms defined in ISO 10303–31

This part of ISO 10303 makes use of the following terms defined in ISO 10303–31:

- preprocessor;
- postprocessor.

3.3 Terms defined in ISO 10303–42

This part of ISO 10303 makes use of the following terms defined in ISO 10303–42:

- arcwise connected;
- bounds;
- boundary;
- boundary representation solid model (B-Rep);
- closed curve;
- closed surface;
- connected;
- constructive solid geometry (CSG);
- coordinate space;
- curve;
- cycle;
- dimensionality;

- domain;
- extent;
- finite;
- geometric coordinate system;
- geometrically founded;
- geometrically related;
- inside;
- interior;
- list;
- model space;
- open curve;
- open surface;
- orientable;
- overlap;
- parameter range;
- parameter space;
- placement coordinate system;
- self–intersect;
- set;
- space dimensionality;
- surface;
- topological sense.

3.4 Terms defined in ISO 10303–44

This part of ISO 10303 makes use of the following terms defined in ISO 10303–44:

- bill-of-material structure (BOM);
- constituent;
- form, fit and function of a product;
- lot.

3.5 Terms defined in ISO 10303–45

This part of ISO 10303 makes use of the following terms defined in ISO 10303–45:

- material;
- material property.

3.6 Terms defined in ISO 10303–46

This part of ISO 10303 makes use of the following terms defined in ISO 10303–46:

- annotation;
- layer;
- picture;
- planar;
- presentation information;
- symbol;
- visualization.

3.7 Terms defined in ISO 10303–47

This part of ISO 10303 makes use of the following terms defined in ISO 10303–47:

- form tolerance;
- profile tolerance;
- toleranced feature.

3.8 Terms defined in ISO 10303–105

This part of ISO 10303 makes use of the following terms in ISO 10303–105:

- base;
- frame;
- joint;
- link;
- mechanism;
- motion;
- pair;
- placement;
- world coordinate system.

3.9 Terms defined in ISO 10303–202

This part of ISO 10303 makes use of the following terms defined in ISO 10303–202:

- application interpreted construct (AIC);
- externally defined;
- predefined.

3.10 Terms defined in ISO 13584–42

This part of ISO 10303 makes use of the following terms defined in ISO 13584–42:

- basic semantic unit (BSU);

3.11 Other definitions

For the purpose of this part of ISO 10303, the following definitions apply.

3.11.1 body in white: Parts of the vehicle body that form the body shell and ensure its stability.

3.11.2 vehicle: Land conveyance for the transportation of goods or passengers.

EXAMPLE 9 – A cart, a lorry, a motor-car, or a bus are examples for a vehicle.

3.12 Abbreviations

For the purpose of this part of ISO 10303, the following abbreviations apply:

C1	surface_condition UoF
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CSG	Constructive Solid Geometry
D1	explicit_draughting UoF
D2	associative_annotation UoF
E1	external_reference_mechanism UoF
FF1	user_defined_feature UoF
FF2	included_feature UoF
FF3	generative_featured_shape UoF
G1	wireframe_model_2d UoF
G2	wireframe_model_3d UoF
G3	connected_surface_model UoF
G4	faceted_b_rep_model UoF
G5	b_rep_model UoF
G6	compound_model UoF
G7	csg_model UoF
G8	geometrically_bounded_surface_model UoF
K1	kinematics UoF
LMC	Least Material condition
MD1	measured_data UoF
MMC	Maximum Material condition

NURBS	Non Uniform Rational B-Spline
P1	geometric_presentation UoF
P2	annotated_presentation UoF
P3	shaded_presentation UoF
PR1	item_property UoF
RGB	Red, green blue colour space
S1	product_management_data UoF
S2	element_structure UoF
S3	item_definition_structure UoF
S4	effectivity UoF
S5	work_management UoF
S6	classification UoF
S7	specification_control UoF
S8	process_plan UoF
T1	dimension_tolerance UoF
T2	geometric_tolerance UoF

4 Information requirements

This clause specifies the information required for core data for automotive mechanical design processes.

The information requirements are specified as a set of units of functionality, application objects, and application assertions. These assertions pertain to individual application objects and to relationships between application objects. The information requirements are defined using the terminology of the subject area of this application protocol.

NOTES

1 – A graphical representation of the information requirements is given in annex G.

2 – The information requirements correspond to those of the activities identified as being within the scope of this application protocol in annex F.

3 – The mapping table specified in 5.1 shows how the integrated resources and application interpreted constructs are used to meet the information requirements of this application protocol.

4.1 Units of functionality

This subclause specifies the units of functionality for the Core Data for Automotive Mechanical Design Processes application protocol.

NOTE 1 – One or more UoFs, grouped together, provide a solution to identified data flows derived from the AAM (see annex F). The grouping of the UoFs is documented in clause 6.

This part of ISO 10303 specifies the following units of functionality:

NOTE 2 – Following the name of each UoF, its abbreviation is shown in brackets. The abbreviations consist of one or two letters and a number. The letters indicate a logical group to which the UoF belongs. The logical groups identified are shown in table 1.

Figure 66 gives an overview for the UoFs, their logical groups, and the interdependencies among the UoFs and the logical groups.

Table 1 – Logical grouping of UoFs

Information type	Abbreviation
surface condition	C
draughting	D
external reference	E
form features	FF
geometry	G
kinematics	K
measured data	MD
property	PR
presentation	P
product structure	S
tolerance	T

- surface_condition (C1);
- explicit_draughting (D1);
- associative_annotation (D2);
- external_reference_mechanism (E1);
- user_defined_feature (FF1);

- included_feature (FF2);
- generative_featured_shape (FF3);
- wireframe_model_2d (G1);
- wireframe_model_3d (G2);
- connected_surface_model (G3);
- faceted_b_rep_model (G4);
- b_rep_model (G5);
- compound_model (G6);
- csg_model (G7);
- geometrically_bounded_surface_model (G8);
- kinematics (K1);
- measured_data (MD1);
- item_property (PR1);
- geometric_presentation (P1);
- annotated_presentation (P2);
- shaded_presentation (P3);
- product_management_data (S1);
- element_structure (S2);
- item_definition_structure (S3);
- effectivity (S4);
- work_management (S5);
- classification (S6);
- specification_control (S7);
- process_plan (S8);

- dimension_tolerance (T1);
- geometric_tolerance (T2).

The units of functionality and a description of the functions that each UoF supports are given below. The application objects included in the UoFs are defined in 4.2.

4.1.1 surface_condition (C1)

This unit of functionality provides the capability to represent surface properties and surface treatments of a material. It covers surface characteristics which may be defined by a designer to achieve the required functionality of a part.

The definition of the presentation of the surface characteristics information is dealt with in the associative_annotation UoF (D2).

The following application objects are used by the surface_condition UoF:

- Coating_layer;
- Coating_layer_relationship;
- Contact_ratio;
- Default_roughness;
- Default_setting_association;
- Hardness;
- Roughness;
- Surface_condition;
- Tactile_appearance;
- Treatment_result;
- Value_with_unit;
- Visual_appearance.

4.1.2 explicit_draughting (D1)

This unit of functionality specifies the representation of annotation elements in an explicit form used in technical drawings. Additionally it defines the organizational structure and management data for technical drawings.

This UoF is based on ISO 10303–201 and establishes the information requirements for the exchange of part definition data in the form of CAD–drawings with explicit annotation. Annotation elements are either defined in a two-dimensional coordinate system, related to the area or view, or in the same coordinate system as the geometry which may be two- or three-dimensional. The annotation elements themselves are always planar.

This UoF also provides for the management of drawing sheets, drawing views, symbols, and dimensioning.

The following application objects are used by the explicit_draughting UoF:

- Annotation_curve;
- Detailed_element;
- Dimension;
- Dimension_callout;
- Dimension_curve;
- Dimension_sequence_pair;
- Drawing;
- Drawing_assignment;
- Drawing_sequence;
- Drawing_sheet;
- Projection_curve;
- View_area_assignment.

4.1.3 associative_annotation (D2)

This unit of functionality specifies the representation of annotation elements in an associative form, i.e., with an association to the annotated geometry.

This unit of functionality is based on ISO 10303–202 and allows the exchange of drawings that maintain an association between the annotation and the geometric shape of a part. Associated annotation elements are either defined in a two-dimensional coordinate system, related to the area or view, or in the same coordinate system as the geometry which may be two- or three-dimensional. The annotation elements themselves are always planar. In the case of associated dimensions the associativity is maintained between the calculated value of the size or distance, the presentation of the value, and the elements from which this value is calculated.

This UoF also allows a reference to tolerance information within geometric dimensions (see UoF T1).

The following application objects are used by the associative_annotation UoF:

- Associated_annotation_symbol;
- Associated_fill_area;
- Associated_leader;
- Associated_projection_curve;
- Detailed_element;
- Dimension_callout;
- Geometric_tolerance_symbol;
- Surface_condition_symbol.

4.1.4 external_reference_mechanism (E1)

This unit of functionality provides a reference mechanism to specify external documents that are associated with objects defined in this part of ISO 10303. These documents may contain data that are not in scope of this part of ISO 10303. This includes conventional non-digital data and digital data in other computer interpretable formats.

EXAMPLE 10 – Technical drawings on paper, standards documents, or handwritten documents are examples of non-digital data referenced by the external reference mechanism.

EXAMPLE 11 – CAD data in a system's native format, text documents in SGML [8], or NC-data in ISO 6983 [6] are examples of digital data referenced by the external reference mechanism.

The following application objects are used by the external_reference_mechanism UoF:

- Archived_document;
- Digital_document;
- Document_relationship;
- Document_type;
- External_document;
- External_document_assignment;
- External_document_version;
- External_library_reference;
- External_model;

- Physical_model.

4.1.5 user_defined_feature (FF1)

This unit of functionality provides the capability to categorize an area of interest in a shape as a feature and to associate parameters as additional information – not intended for shape generation – with the identified aspect of the shape. Using this mechanism multiple aspects of the shape that are identified as features may be combined into a compound general feature.

The following application objects are used by the user_defined_feature UoF:

- Feature_occurrence;
- Feature_parameter;
- General_compound_feature;
- General_feature;
- General_feature_parameter;
- Included_feature;
- Value_with_unit.

4.1.6 included_feature (FF2)

This unit of functionality provides the capability to identify areas of interest on an already designed shape as a particular kind of feature whose type and defining parameters are specified in this part of ISO 10303.

NOTE 1 – These features usually have a certain meaning to the user, such as a correspondence to a manufacturing process operation managed or manageable by the user.

The feature definitions in this part of ISO 10303 distinguishes between form features for sheet type of parts or tools and form features for solid type of parts or tools. Among these features are hole, pocket, boss, slot, rib, thread, bead and joggle as well as transition features such as fillet, edge round, and chamfer. A feature replication mechanism with various patterns as well as a compound feature mechanism are also included.

Within this UoF the specification of parameters on these features does not mean that a variation of these parameters is intended in order to change the shape of the part the feature is applied to.

NOTE 2 – The parameters may be used to specify the areas of interest in a more appropriate way, e.g., to support manufacturing planning.

The following application objects are used by the included_feature UoF:

- Angle_specification;

- Angle_taper;
- Bead_end_type;
- Bead_feature;
- Boss_feature;
- Boss_top_condition;
- Catalog_thread;
- Chamfer;
- Chamfer_angle;
- Chamfer_definition;
- Chamfer_offset;
- Circular_boss;
- Circular_closed_profile;
- Circular_offset_pattern;
- Circular_omit_pattern;
- Circular_path;
- Circular_pattern;
- Closed_profile;
- Closed_profile_usage;
- Complete_circular_path;
- Compound_feature_in_panel;
- Compound_feature_in_panel_element;
- Compound_feature_in_panel_element_relationship;
- Compound_feature_in_solid;
- Compound_feature_in_solid_element;

- Compound_feature_in_solid_element_relationship;
- Conical_bottom;
- Constant_radius_edge_round;
- Constant_radius_fillet;
- Default_chamfer;
- Default_draft_angle;
- Default_edge_round;
- Default_fillet;
- Default_setting_association;
- Defined_thread;
- Diameter_taper;
- Directed_taper;
- Edge_round;
- Face_termination;
- Feature_definition;
- Feature_occurrence;
- Fillet;
- Flat_bottom;
- Flat_slot_end_type;
- Flat_with_chamfer_bottom;
- Flat_with_radius_bottom;
- Form_feature_in_panel;
- General_boss;
- General_bottom_condition;

- General_closed_profile;
- General_open_profile;
- General_path;
- General_pattern;
- General_pocket_in_solid;
- General_top_condition;
- Hole_bottom_condition;
- Hole_feature_in_panel;
- Joggle_feature;
- Joggle_termination;
- Linear_bead_end_type;
- Linear_path;
- Locator_feature;
- Ngon_profile;
- Open_bead_end_type;
- Open_profile;
- Open_slot_end_type;
- Panel_hole_with_barring;
- Partial_circular_path;
- Partial_circular_profile;
- Path;
- Planar_bottom_condition;
- Planar_top_condition;
- Pocket_bottom_condition;

- Pocket_feature_in_solid;
- Process_feature_in_panel;
- Process_feature_in_solid;
- Profile;
- Radiused_bead_end_type;
- Radiused_slot_end_type;
- Rectangular_boss;
- Rectangular_closed_pocket_in_solid;
- Rectangular_closed_profile;
- Rectangular_offset_pattern;
- Rectangular_omit_pattern;
- Rectangular_open_pocket_in_solid;
- Rectangular_pattern;
- Replicate_base_assignment;
- Replicate_feature;
- Rib_feature;
- Round_hole;
- Rounded_u_profile;
- Slot_end_type;
- Slot_feature;
- Spherical_bottom;
- Square_u_profile;
- Tee_profile;
- Thread_feature;

- Thread_specification;
- Through_bottom_condition;
- Through_hole_condition;
- Transition_feature;
- Value_termination;
- Vee_profile;
- Woodruff_slot_end_type.

4.1.7 generative_featured_shape (FF3)

This unit of functionality provides the capability to use a subset of those features specified within the included_feature UoF in a way that the application of the features on a base shape results in a modified shape. Within this UoF the identification of feature parameters is related to the intent to study the consequences of a variation of these parameters on the resulting shape.

Additionally this UoF supports the definition of sequences of 'operations', i.e. the application of one or more features to a base shape with the possibility to represent the resulting shapes at the various intermediate stages.

The following application objects are used by the generative_featured_shape UoF:

- Feature_occurrence;
- Feature_placement;
- Featured_shape;
- Placed_feature.

4.1.8 wireframe_model_2d (G1)

This unit of functionality provides the capability to represent geometry as a two-dimensional wireframe model, i.e., of two-dimensional points and curves. The difference between 3D and 2D geometric elements is specified by the presence or absence of a third coordinate value, respectively.

Composite curves for the geometric representation of paths and loops are included in this UoF. The types of points and curves excluded from this UoF are point on surface and curve on surface. All unbounded curves in this UoF are explicitly trimmed unless they are closed.

Surface data and all topological information are outside the scope of this UoF.

The following application objects are used by the wireframe_model_2d UoF:

- B_spline_curve;
- Cartesian_point;
- Circle;
- Composite_curve;
- Constructive_geometric_element;
- Constructive_geometry;
- Constructive_geometry_association;
- Curve;
- Curve_set_2d;
- Detailed_element;
- Detailed_geometric_model_element;
- Detailed_model_element;
- Ellipse;
- Geometric_model;
- Geometric_model_relationship;
- Geometric_model_version;
- Hyperbola;
- Line;
- Offset_curve;
- Parabola;
- Point;
- Point_on_curve;
- Polyline;
- Trimmed_curve;

- Wireframe_model_2d.

4.1.9 wireframe_model_3d (G2)

This unit of functionality provides the capability to represent geometry as a three-dimensional geometrically bounded wireframe model. The difference between 3D and 2D geometric elements is specified by the presence or absence of a third coordinate value, respectively. All application objects of the wireframe_model_2d UoF (G1) are included for the 3D case in the wireframe_model_3d UoF (G2).

Composite curves for the geometric representation of paths and loops are included in this UoF. The types of points and curves excluded from this UoF are point on surface and curve on surface. All unbounded curves in this UoF are explicitly trimmed unless they are closed.

Surface data and all topological information are outside the scope of this UoF.

The following application objects are used by the wireframe_model_3d UoF:

- B_spline_curve;
- Cartesian_point;
- Circle;
- Composite_curve;
- Constructive_geometric_element;
- Constructive_geometry;
- Constructive_geometry_association;
- Curve;
- Curve_set_3d;
- Detailed_element;
- Detailed_geometric_model_element;
- Detailed_model_element;
- Ellipse;
- Geometric_model;
- Geometric_model_relationship;
- Geometric_model_version;

- Hybrid_geometric_model_3d;
- Hyperbola;
- Line;
- Offset_curve;
- Parabola;
- Point;
- Point_on_curve;
- Polyline;
- Trimmed_curve;
- Wireframe_model_3d.

4.1.10 connected_surface_model (G3)

This unit of functionality provides the capability to represent geometry as a surface model that is topologically bounded.

Topological relationships are modeled explicitly by using objects, i.e., face, loop, edge, and vertex. The bounded surfaces in this UoF are connected. A model shall be manifold. A facility to associate transition information with a pair of adjacent faces or with a pair of adjacent edges is provided. Trimming and composition of curves and surfaces are represented by applying topological objects, i.e., vertex, edge, loop, and face.

This unit of functionality contains the geometrical representation of points, curves, and surfaces of the wireframe_model_3D UoF (G2) and of the geometrically_bounded_surface_model UoF (G8). The connected_surface_model UoF has the additional functionality of representing points and curves in the parameter space of curves and surfaces, respectively.

EXAMPLE 12 – Examples for curves represented in the parameter space of surfaces are intersection curves or seam curves.

The following application objects are used by the connected_surface_model UoF:

- B_spline_curve;
- B_spline_surface;
- Cartesian_point;
- Circle;
- Conical_surface;

- Constructive_geometric_element;
- Constructive_geometry;
- Constructive_geometry_association;
- Curve;
- Curve_on_surface;
- Cylindrical_surface;
- Detailed_element;
- Detailed_geometric_model_element;
- Detailed_model_element;
- Edge_loop;
- Edge_transition;
- Ellipse;
- Face;
- Face_bound;
- Face_transition;
- Geometric_model;
- Geometric_model_relationship;
- Geometric_model_version;
- Hybrid_geometric_model_3d;
- Hyperbola;
- Line;
- Offset_surface;
- Oriented_edge_curve;
- Parabola;

- Plane;
- Point;
- Point_on_curve;
- Point_on_surface;
- Polyline;
- Shell;
- Spherical_surface;
- Surface;
- Surface_model;
- Surface_of_linear_extrusion;
- Surface_of_revolution;
- Toroidal_surface;
- Vertex_loop;
- Vertex_point.

4.1.11 faceted_b_rep_model (G4)

This unit of functionality provides the capability to represent geometry as a B-rep model with planar faces and implicit topology.

The faceted B-rep model is represented by an outer shell and an arbitrary number of internal voids, each of which is represented by a closed shell. A requirement, included to provide internal consistency checks, is that each face shall be associated with a plane surface. All the face boundaries are implicitly defined by poly loops. One of these poly loops shall be designated the outer boundary. The edges are all straight lines but are not explicitly defined. The edge information can be deduced from the poly loop definitions.

This UoF shares the basic geometric definitions of point, line, and plane with the UoFs G3 and G5. It also shares the definition of a manifold solid B-rep and the topological concepts of closed shell and face with UoF G5.

NOTE – A faceted B-rep model, as defined in this UoF would not be considered to be a valid example of a B-rep model in the UoF G5; it could however be converted to such a model by explicitly creating lines, edges and vertices and converting the face bounding poly loops in UoF G4 to edge loops as required by UoF G5.

The following application objects are used by the faceted_b_rep_model UoF:

- Cartesian_point;
- Constructive_geometric_element;
- Constructive_geometry;
- Constructive_geometry_association;
- Detailed_element;
- Detailed_geometric_model_element;
- Detailed_model_element;
- Faceted_b_rep;
- Faceted_b_rep_model;
- Faceted_closed_shell;
- Faceted_face;
- Geometric_model;
- Geometric_model_relationship;
- Geometric_model_version;
- Hybrid_geometric_model_3d;
- Plane;
- Point;
- Poly_loop.

4.1.12 b_rep_model (G5)

This unit of functionality provides the capability to represent geometry as B-rep model with a consistent topological structure and directly defined geometry for faces and edges.

The B-rep model is represented by one or more B-rep solids. Each B-rep solid consists of a single closed shell, which represents its exterior, and an arbitrary number of closed shells corresponding to internal voids. Each shell consists of a collection of faces. Each face is defined by a surface and by loops of edges bounding the surface. The edges meet at vertices and have their geometry

defined by curves. The geometry of the curves and surfaces can be complex but is required to be directly defined. Constraints ensure that the B-rep model is defined in the simplest and most direct manner consistent with the requirement to provide a complete geometric representation.

All the geometry defined in this UoF is shared with the connected_surface_model UoF (G3), the topological definitions of vertex, edge, path, loop, and face are also common between the UoFs G3 and G5.

NOTE – Any B-rep model defined in UoF G5 could be represented in UoF G3 as a collection of faces with no loss of geometric precision but losing the information of a solid model, e.g., inner and outer domain of the B-rep solid.

The following application objects are used by the b_rep_model UoF:

- B_rep;
- B_rep_model;
- B_spline_curve;
- B_spline_surface;
- Cartesian_point;
- Circle;
- Closed_shell;
- Conical_surface;
- Constructive_geometric_element;
- Constructive_geometry;
- Constructive_geometry_association;
- Curve;
- Curve_on_surface;
- Cylindrical_surface;
- Detailed_element;
- Detailed_geometric_model_element;
- Detailed_model_element;
- Edge_loop;

- Edge_transition;
- Ellipse;
- Face;
- Face_bound;
- Face_transition;
- Geometric_model;
- Geometric_model_relationship;
- Geometric_model_version;
- Hybrid_geometric_model_3d;
- Hyperbola;
- Line;
- Oriented_edge_curve;
- Parabola;
- Plane;
- Point;
- Polyline;
- Spherical_surface;
- Surface;
- Surface_of_linear_extrusion;
- Surface_of_revolution;
- Toroidal_surface;
- Vertex_loop;
- Vertex_point.

4.1.13 compound_model (G6)

This unit of functionality provides the capability to represent geometry as a topologically connected geometric model consisting of solid, surface, or wireframe data. This geometric model deals with mixed dimensionality (solids, surfaces, and wireframes) and non-manifold topology.

Only shape models with complete topological description, i.e., in boundary representation are supported. Since the topology is not restricted to be manifold, the models can have more than two faces sharing one edge and have faces shared by two adjacent shells. A compound representation collects several geometric items of the same or of different type and joins them into one topologically connected model.

NOTE – This UoF should be used wherever portions of the model have to be distinguished.

EXAMPLE 13 – Examples for distinguishing portions of the model are material zones with different heat treatment, structures welded together, meshing macros for finite element analysis, and shape representation of varying dimensionality, e.g., surfaces and solids.

This UoF includes all application objects of UoF b_rep_model (G5) and connected_surface_model (G3).

The following application objects are used by the compound_model UoF:

- B_spline_curve;
- B_spline_surface;
- Cartesian_point;
- Circle;
- Compound_model;
- Conical_surface;
- Constructive_geometric_element;
- Constructive_geometry;
- Constructive_geometry_association;
- Curve;
- Curve_on_surface;
- Cylindrical_surface;
- Detailed_element;
- Detailed_geometric_model_element;

- Detailed_model_element;
- Edge_loop;
- Edge_transition;
- Ellipse;
- Face;
- Face_bound;
- Face_transition;
- Geometric_model;
- Geometric_model_relationship;
- Geometric_model_version;
- Hybrid_geometric_model_3d;
- Hyperbola;
- Line;
- Offset_surface;
- Oriented_edge_curve;
- Parabola;
- Plane;
- Point;
- Point_on_curve;
- Point_on_surface;
- Polyline;
- Spherical_surface;
- Surface;
- Surface_of_linear_extrusion;

- Surface_of_revolution;
- Topologically_connected_set;
- Toroidal_surface;
- Vertex_loop;
- Vertex_point.

4.1.14 csg_model (G7)

This unit of functionality provides the capability to represent geometry as a solid model using Constructive Solid Geometry (CSG).

A CSG shape is the evaluation of a CSG tree which has solid objects at the leaves and boolean operations at the nodes. This model provides geometric functionality for building solids using boolean operations on predefined simple parameterized solid shapes, manifold solid B-rep shapes in faceted or exact form, and half space solids.

The following application objects are used by the csg_model UoF:

- B_rep;
- B_rep_model;
- B_spline_curve;
- B_spline_surface;
- Block;
- Boolean_result;
- Cartesian_point;
- Circle;
- Closed_shell;
- Conical_surface;
- Constructive_geometric_element;
- Constructive_geometry;
- Constructive_geometry_association;
- Csg_model;

- Csg_primitive;
- Curve;
- Curve_on_surface;
- Cylindrical_surface;
- Detailed_element;
- Detailed_geometric_model_element;
- Detailed_model_element;
- Edge_loop;
- Edge_transition;
- Ellipse;
- Face;
- Face_bound;
- Face_transition;
- Faceted_b_rep;
- Faceted_b_rep_model;
- Faceted_closed_shell;
- Faceted_face;
- Geometric_model;
- Geometric_model_relationship;
- Geometric_model_version;
- Half_space_solid;
- Hybrid_geometric_model_3d;
- Hyperbola;
- Line;

- Oriented_edge_curve;
- Parabola;
- Plane;
- Point;
- Poly_loop;
- Polyline;
- Right_angular_wedge;
- Right_circular_cone;
- Right_circular_cylinder;
- Solid_of_linear_extrusion;
- Solid_of_revolution;
- Solid_replica;
- Sphere;
- Spherical_surface;
- Surface;
- Surface_of_linear_extrusion;
- Surface_of_revolution;
- Swept_face_solid;
- Toroidal_surface;
- Torus;
- Vertex_loop;
- Vertex_point.

4.1.15 geometrically_bounded_surface_model (G8)

This unit of functionality provides the capability to represent geometry as a surface model that is geometrically bounded.

This UoF provides basic geometric constructs, such as three-dimensional points, curves, and surfaces. It includes both elementary and sculptured surfaces. Points and curves can be represented in the parameter space of curves or surfaces, respectively, by objects such as point on curve and parametric curve. The replication of the geometric objects is not included in this UoF.

The basic geometric constructs are used for the definition of complex geometric objects, i.e., geometrically bounded surfaces without connectivity. They are also used within the connected_surface_model UoF (G3) and b_rep_model UoF (G5) for the definition of topologically bounded surfaces with topological connections.

NOTE – This UoF does not contain any topological information. Therefore the geometrically bounded surfaces cannot be connected by topology.

The following application objects are used by the geometrically_bounded_surface_model UoF:

- B_spline_curve;
- B_spline_surface;
- Cartesian_point;
- Circle;
- Composite_curve;
- Conical_surface;
- Constructive_geometric_element;
- Constructive_geometry;
- Constructive_geometry_association;
- Curve;
- Curve_on_surface;
- Cylindrical_surface;
- Ellipse;
- Geometric_model;
- Geometric_model_relationship;

- Geometric_model_version;
- Geometric_set_3d;
- Geometrically_bounded_surface_model;
- Hyperbola;
- Line;
- Offset_curve;
- Offset_surface;
- Parabola;
- Plane;
- Point;
- Point_on_curve;
- Point_on_surface;
- Polyline;
- Spherical_surface;
- Surface;
- Surface_of_linear_extrusion;
- Surface_of_revolution;
- Toroidal_surface;
- Trimmed_curve;
- Trimmed_surface.

4.1.16 kinematics (K1)

This unit of functionality provides the capability to represent kinematic aspects of a mechanical product as required for the communication between CAD systems and kinematic analysis systems, including robotic simulation systems, as well as between dissimilar kinematic analysis systems.

A mechanism as described using this UoF is regarded as being composed of rigid links mating in the following types of joints:

- cylindrical pair;
- gear pair;
- planar pair;
- point on surface pair;
- prismatic pair;
- rack and pinion pair;
- revolute pair;
- rolling curve pair;
- rolling surface pair;
- screw pair;
- sliding curve pair;
- sliding surface pair;
- spherical pair;
- universal pair.

The mechanism may contain open or closed kinematic chains. This information is implicitly defined by the topology of the kinematic structure. A mechanism may be connected to another mechanism. Both planar and 3D mechanisms can be defined. The rigid links may be described by a 3D geometric shape description. A link may have additional frames attached to it, to allow application specific kinematic forward analysis to be performed.

The following application objects are used by the kinematics UoF:

- Actuator;
- Ground_representation;
- Initial_configuration;
- Interpolated_configuration_sequence;
- Kinematic_configuration_definition;
- Kinematic_configuration_interpolation;

- Kinematic_joint;
- Kinematic_link;
- Kinematic_link_representation;
- Kinematic_mechanism;
- Kinematic_pair;
- Kinematic_structure;
- Kinematic_transformation_result;
- Pair_value;
- Path_approximation;
- Simple_pair_range;
- Simulation_information;
- Tool_attachment_point_frame;
- Tool_center_point_frame;
- Transformation_to_reference_frame.

4.1.17 measured_data (MD1)

This unit of functionality specifies the representation of scanned data resulting from a measurement process of a physical part. These data are represented by a sequence of measured points, potentially with associated vector information for the measured surface normal direction. This data representation may also be used for manufacturing processes, e.g., to represent tool paths for NC programming with the vector information indicating the tool direction.

The following application objects are used by the measured_data UoF:

- Point_direction;
- Point_direction_model;
- Point_direction_sequence;
- Point_sequence.

4.1.18 item_property (PR1)

This unit of functionality specifies various properties for a part, such as material, recyclability, mass, and general properties.

The information about recyclability is required to describe whether and how material or component parts may be reused. The costs as planned or calculated for a part as well as a quality measure property to keep track of the results of some processes, e.g., prototype building, may be specified. The mass is required to express the weight of a part. Mass properties that are rotation and translation invariant such as the centre of mass and the moments of inertia may also be specified.

The general properties are required to define property data which are not covered by one of the other UoFs, and, due to their variety, cannot be specified all by explicit application objects.

> EXAMPLE 14 – Noise development of an engine, maintenance intervals of parts, and heat or noise absorption of insulating parts between the engine and passenger compartment are examples for general properties.

The following application objects are used by the item_property UoF:

- Centre_of_mass;
- Data_environment;
- Default_setting_association;
- Default_thickness;
- General_item_property;
- Item_cost_property;
- Item_property;
- Item_quality_property;
- Mass;
- Material;
- Material_property;
- Material_property_representation;
- Material_side_specification;
- Moments_of_inertia;
- Property;

- Property_change;
- Property_relationship;
- Property_value;
- Recyclability;
- Shape_dependent_property;
- Thickness;
- Value_with_unit.

4.1.19 geometric_presentation (P1)

This unit of functionality specifies the presentation of point, vector, curve, and surface geometry with attributes, such as colour, line thickness, or line font. The association of these attributes to geometric elements is defined either directly or through a context.

EXAMPLE 15 – The following are examples for a context: drawing sheet, layer, group, model view, template, view, or view area.

The curve appearance is restricted to unscaled curve fonts only. The viewing mechanism with one or more views onto the geometric shape model is also provided.

The following application objects are used by the geometric_presentation UoF:

- Appearance_assignment;
- Context_dependent_appearance_assignment;
- Curve_appearance;
- Model_image;
- Point_appearance;
- Styled_element;
- Styled_geometric_model;
- Styled_model;
- Styled_model_element;
- Styled_model_relationship;
- Styled_model_version;

- Surface_appearance;
- Surface_appearance_element;
- Vector_appearance;
- View;
- View_area.

4.1.20 annotated_presentation (P2)

This unit of functionality specifies the presentation of annotations and a view layout with multiple views, including annotations. The curve appearance is unrestricted in this UoF.

EXAMPLE 16 – Sketches can be represented with this UoF.

The organizational definitions of technical drawings is not included in this UoF but in the Explicit_draughting UoF.

The following application objects are used by the annotated_presentation UoF:

- Annotation_curve;
- Annotation_element;
- Annotation_model;
- Annotation_subfigure;
- Annotation_symbol;
- Area_placed_annotation;
- Detailed_element;
- Draughting_callout;
- Fill_area;
- Hatching;
- Leader;
- Solid_fill_area;
- Text;
- Text_appearance;

- Tiling;
- View_area_relationship;
- View_placed_annotation;
- View_relationship.

4.1.21 shaded_presentation (P3)

This unit of functionality specifies advanced presentation capabilities for three-dimensional geometry. This includes hidden line and hidden surface presentation and definitions which enable the creation of shaded images, such as light source definitions. Enhanced surface appearance definitions are also included.

EXAMPLE 17 – Examples of enhanced surface appearance definitions are transparency and reflection rates of surfaces.

The following application objects are used by the shaded_presentation UoF:

- Curve_appearance;
- Hidden_element_handling;
- Light_source;
- Surface_appearance;
- Surface_appearance_element.

4.1.22 product_management_data (S1)

This unit of functionality provides the capability to represent product management information. It includes information about items that are either raw materials, parts, or tools, about versions and views of items, about persons and dates, and about information resulting from release and approval processes.

Additionally this unit of functionality provides the capability to represent descriptive information about objects in different languages and to assign alias identifiers to objects.

The following application objects are used by the product_management_data UoF:

- Alias_identification;
- Application_context;
- Approval;
- Approval_relationship;

- Approval_status;
- Date_and_person_assignment;
- Date_and_person_or_organization;
- Date_time;
- Date_time_assignment;
- Design_discipline_item_definition;
- Item;
- Item_classification;
- Item_version;
- Multi_language_string;
- Organization;
- Person;
- Person_in_organization;
- Person_or_organization_assignment;
- Security_classification;
- Security_level;
- Specific_classification;
- String_with_language.

4.1.23 element_structure (S2)

This unit of functionality provides the capability to associate items with shape or to identify aspects of the shape. Additionally this UoF contains elements that are needed to build up structures of data elements.

The data elements that can be structured are geometric elements, annotation elements, and kinematic elements. The supported structures are groups, layers, and templates for multiple instances of the same definition.

Group structures and template structures may be nested. Layers can not be nested but may be related.

The following application objects are used by the element_structure UoF:

- Accuracy;
- Axis_placement;
- Cartesian_coordinate_space;
- Cartesian_coordinate_space_2d;
- Cartesian_coordinate_space_3d;
- Geometric_model_relationship_with_transformation;
- Group;
- Group_relationship;
- Item_shape;
- Layer;
- Model_change;
- Shape_aspect;
- Shape_aspect_relationship;
- Shape_description_association;
- Structuring_method_hierarchy;
- Template_instance;
- Transformation;
- Transformation_2d;
- Transformation_3d.

4.1.24 item_definition_structure (S3)

This unit of functionality provides the capability to represent relationships between items to build up various kinds of structures.

The structures are defined for design discipline item definitions, for item instances that are particular occurrences of design discipline item definitions or between design discipline item definitions and item instances.

Among these structures is a hierarchical assembly structure to represent the relationships among constituents and the assembly.

The following application objects are used by the item_definition_structure UoF:

- Assembly_component_relationship;
- Assembly_definition;
- Assembly_substitute_relationship;
- General_item_definition_instance_relationship;
- General_item_definition_relationship;
- General_item_instance_relationship;
- Geometrical_relationship;
- Item_definition_instance_relationship;
- Item_definition_relationship;
- Item_instance;
- Item_instance_relationship;
- Next_higher_assembly;
- Promissory_usage;
- Quantified_instance;
- Selected_instance;
- Single_instance;
- Value_with_unit.

4.1.25 effectivity (S4)

This unit of functionality provides the capability to represent information concerning the validity of data.

The validity of data can be expressed by effectivities that specify time ranges within which data may be used and by retention periods that specify how long data have to kept and when they may or have to be deleted.

Both concepts can make use of explicit dates or of dates expressed by events in order to represent the relevant points in time.

The following application objects are used by the effectivity UoF:

- Effectivity;
- Effectivity_relationship;
- Event_reference;
- Retention_period;
- Value_with_unit.

4.1.26 work_management (S5)

This unit of functionality provides the capability to represent activity, project, and contract related information.

Activities may be initiated by work requests and may be authorized by work orders. Activities may result in changes of models or properties; such changes can also be represented in this UoF.

The following application objects are used by the work_management UoF:

- Activity;
- Activity_element;
- Activity_method;
- Activity_relationship;
- Alternate_item_relationship;
- Change;
- Contract;
- Element_delivery;
- Item_version_relationship;
- Organization_in_contract;
- Organization_relationship;
- Project;
- Project_relationship;

- Simultaneous_activity;
- Value_with_unit;
- Work_order;
- Work_request.

4.1.27 classification (S6)

This unit of functionality provides the capability to classify items into specific categories.

EXAMPLE 18 – Examples are the classification as raw material, parts, or tools, or as safety or in-process items.

Additionally items may be classified by their attributes.

EXAMPLE 19 – Examples for categories of attribute based classification are screws, bolts, nuts, shafts, or brackets.

The following application objects are used by the classification UoF:

- Classification_attribute;
- Classification_system;
- General_classification;
- General_classification_hierarchy;
- Item_classification;
- Specific_classification.

4.1.28 specification_control (S7)

This unit of functionality provides the capability to describe automotive products with a large number of variants.

EXAMPLE 20 – Examples for automotive products are passenger cars, trucks, busses, or engines or other components of these products.

Because of the large number of variants there is no unique identification for each of the variants that could be produced. The main concepts that are used to handle this large number of variants are the following:

- product classes are used to identify sets of similar products to be offered to the market;
- specifications are used to describe characteristics of the products;
- specification categories are used to group similar characteristics of the products;

– specification expressions are used to control the usage of a part within a product and to represent conditions for product classes;

– product functions are used to describe the functional requirements for products and components of products;

– product components are used to describe the common decomposition structure of all products of a product class;

– item solutions are used to describe the variants for a product component;

– item instances are used to identify elements of an item solution.

The following application objects are used by the specification_control UoF:

– Class_category_association;

– Class_condition_association;

– Class_function_association;

– Class_inclusion_association;

– Class_specification_association;

– Collected_item_association;

– Collection_definition;

– Configuration;

– Dated_configuration;

– Descriptive_specification;

– Design_constraint;

– Design_constraint_association;

– Design_constraint_relationship;

– Design_constraint_version;

– Final_item_solution;

– Function_component_association;

– Instance_placement;

- Item_solution;
- Item_solution_relationship;
- Lot_configuration;
- Physical_instance;
- Physical_instance_usage;
- Process_operation;
- Product_class;
- Product_class_hierarchy;
- Product_component;
- Product_component_relationship;
- Product_function;
- Product_function_relationship;
- Product_specification;
- Replaced_definition_relationship;
- Replaced_usage_relationship;
- Serial_configuration;
- Solution_certification;
- Solution_instance_association;
- Specification;
- Specification_category;
- Specification_expression;
- Specification_inclusion;
- Specified_higher_usage;
- Supplier_solution;

- Technical_solution.

4.1.29 process_plan (S8)

This unit of functionality provides the capability to represent process related data. This includes process plans, versions of process plans with version tracking, process operations and properties of processes.

A process plan is decomposed into one or more process operations. Process plans and process operations establish relationships among raw materials, in-process parts, and final parts, as well as the relationship between the parts and the tools used to manufacture them.

Additionally, the representation of the connection of parts in various kinds of matings is part of this UoF.

The following application objects are used by the process_plan UoF:

- Exclusive_process_operation;
- General_process_property;
- Make_from_relationship;
- Mated_item_association;
- Mated_item_relationship;
- Mating_definition;
- Process_cost_property;
- Process_operation;
- Process_operation_input_or_output;
- Process_operation_relationship;
- Process_operation_resource;
- Process_operation_resource_assignment;
- Process_plan;
- Process_plan_operation_assignment;
- Process_plan_relationship;
- Process_plan_version;

- Process_property;
- Process_quality_property;
- Process_state;
- Same_time_machining_relationship;
- Simultaneous_process_operation;
- Tool_part_relationship.

4.1.30 dimension_tolerance (T1)

This unit of functionality specifies the representation of geometric dimensions and of tolerances limiting geometric dimensions. Tolerances for geometric dimensions are characterized by plus-minus-tolerances, limits and fits, dimension ranges, or dimension limits and describe the allowable deviation range.

The definition of the presentation of the Tolerance information is dealt with in the associative_-annotation UoF (D2).

The following application objects are used by the dimension_tolerance UoF:

- Angular_location_dimension;
- Angular_size_dimension;
- Curved_distance_dimension;
- Default_setting_association;
- Derived_geometry;
- Dimension_tolerance;
- General_tolerance_table;
- General_tolerances;
- Geometric_dimension;
- Limits_and_fits;
- Linear_distance_dimension;
- Location_dimension;
- Plus_minus_tolerance;

- Plus_minus_toleranced_datum;
- Significant_number_of_digits;
- Size_dimension;
- Tolerance_table_cell;
- Upper_lower_limit;
- Upper_lower_toleranced_datum;
- Value_limit;
- Value_range.

4.1.31 geometric_tolerance (T2)

This unit of functionality specifies the representation of geometric tolerances with a datum reference, such as parallelism or perpendicularity, and also geometric tolerances without a datum reference, such as straightness or flatness.

This UoF allows the definition of up to three datum references that are either single datums, compound datums, or datum targets and additionally the specification of tolerance zones.

The definition of the presentation of the Tolerance information is dealt with in the associative_-annotation UoF (D2).

The following application objects are used by the geometric_tolerance UoF:

- Angularity_tolerance;
- Circular_runout_tolerance;
- Circularity_tolerance;
- Compound_datum;
- Concentricity_tolerance;
- Cylindricity_tolerance;
- Datum;
- Datum_feature;
- Datum_target;
- Datum_target_set;

- Derived_geometry;
- Flatness_tolerance;
- Geometric_tolerance;
- Geometric_tolerance_precedence;
- Linear_profile_tolerance;
- Material_condition;
- Parallelism_tolerance;
- Perpendicularity_tolerance;
- Placed_target;
- Position_tolerance;
- Projection;
- Reference_plane;
- Straightness_tolerance;
- Surface_profile_tolerance;
- Symmetry_tolerance;
- Target_area;
- Target_circle;
- Target_line;
- Target_point;
- Target_rectangle;
- Tolerance_zone;
- Tolerance_zone_definition;
- Total_runout_tolerance.

4.2 Application objects

This subclause specifies the application objects for the Core Data for Automotive Mechanical Design Processes application protocol. Each application object is an atomic element that embodies a unique application concept and contains attributes specifying the data elements of the object. The application objects and their definitions are given below.

NOTE 1 – The following conventions apply for the definition text:

– lower case terms refer to the concept known in the application environment, e.g., 'circle' refers to the concept of a round plane figure whose circumference is everywhere equidistant from its centre;

– upper case terms refer to the application objects as defined in this part of ISO 10303, e.g., 'Circle' refers to the application object as defined in the geometric UoFs.

The attribute names 'id', 'name', and 'description' imply the following meanings:

- id: a computer interpretable identifier;
- name: human readable information;
- description: descriptive text.

NOTE 2 – The id is not a system generated identifier used by an implementation to distinguish between the various objects but is an identifier that has a certain meaning in an application and that may be meaningful to a human being as well.

In this clause a permissive list approach for attributes with suffix '_type' or with name 'role' is used consistently, i.e. there is a list of standardized values with the specific semantics defined. Other non-standardized values are allowed as well.

NOTE 3 – Especially many relationship objects are defined in a general way where the attribute 'relation_type' defines the standardized usages with a definition of the meaning of the other attributes, like 'relating' and 'related'.

4.2.1 Accuracy

An Accuracy is the information about the geometrical accuracy of the product data contained in a model.

The data associated with an Accuracy are the following:

- angular_accuracy_value;
- curvature_accuracy_value;
- distance_accuracy_value;
- is_defined_for.

4.2.1.1 angular_accuracy_value

The angular_accuracy_value specifies the maximum value for the absolute angle between two curve tangents or two surface normals for which the creating system assumes curve tangents or surface normals being identical.

NOTE – The angular_accuracy_value can be used to determine the accuracy range for tangent continuous curve or surface connections.

The angular_accuracy_value need not be specified for a particular Accuracy.

4.2.1.2 curvature_accuracy_value

The curvature_accuracy_value specifies the maximum value for the term

$$\frac{2\|R_1 - R_2\|}{\|R_1\| + \|R_2\|}$$

for which the creating system assumes the two radii of curvature R_1 and R_2 to be identical.

NOTE – Terms surrounded by the || symbols represent the absolute values of these terms.

The curvature_accuracy_value is used to determine the accuracy range for curvature continuous curve or surface connections.

The curvature_accuracy_value need not be specified for a particular Accuracy.

4.2.1.3 distance_accuracy_value

The distance_accuracy_value specifies the allowable maximum value of all possible minimal distances between a point of the concerned geometric element and all points that are assumed to be identical.

The distance_accuracy_value defined for a Geometric_model (see 4.2.211) is valid for all geometric elements of the Geometric_model.

EXAMPLE 21 – In the case of a Vertex_point (see 4.2.489) the distance_accuracy_value is the allowable distance between a point and a point that is assumed to be identical. In the case of an Oriented_edge_curve (see 4.2.297) the distance_accuracy_value is the allowable maximum value of all possible minimal distances between a point on an Oriented_edge_curve and an Oriented_edge_curve that is assumed to be identical.

4.2.1.4 is_defined_for

The is_defined_for specifies the description that the Accuracy is associated to.

See 4.3.1, 4.3.2, 4.3.3, and 4.3.4 for the application assertions.

There shall be at least one object that the Accuracy is_defined_for.

4.2.2 Activity

An Activity is the fact of achieving or accomplishing an action.

The data associated with an Activity are the following:

- activity_type;
- actual_end_date;
- actual_start_date;
- chosen_method;
- concerned_organization;
- description;
- id;
- planned_end_date;
- planned_start_date;
- requestor;
- resolved_request;
- status;
- supplying_organization.

4.2.2.1 activity_type

The activity_type specifies the kind of the Activity. Where applicable the following values shall be used:

- 'amendment': An activity to add information to product data;
- 'analysis': An activity to determine the behaviour of an item version under certain physical circumstances;
- 'cancellation': An activity to delete an item from the bill of material or to cancel the whole bill of material;
- 'delivery change': An activity to change an element delivery;
- 'design change': An activity to change the design or data like geometry, master data or properties of an item or an assembly;
- 'design': An activity concerning the development of a design of an item or a product;
- 'mock-up creation': An activity to create an experimental model or replica of an item;

– 'prototype building': An activity to manufacture a preliminary version of an item or product;

– 'rectification': An activity to correct the data, documentation or structure belonging to an item;

– 'restructuring': An activity to create a new structure or position within a bill of material without changing the data of the items;

– 'spare part creation': An activity to design a spare part or to classify an item as a spare part;

– 'stop notice': An activity to stop the manufacturing process of an item;

– 'testing': An activity to test an item or a product.

4.2.2.2 actual_end_date

The actual_end_date specifies the date when the Activity was actually finished.

See 4.3.7 for the application assertion.

The actual_end_date need not be specified for a particular Activity.

4.2.2.3 actual_start_date

The actual_start_date specifies the date when the Activity was actually started.

See 4.3.7 for the application assertion.

The actual_start_date need not be specified for a particular Activity.

4.2.2.4 chosen_method

The chosen_method specifies the Activity_method (see 4.2.4) used to carry out the Activity.

See 4.3.5 for the application assertion.

The chosen_method need not be specified for a particular Activity.

4.2.2.5 concerned_organization

The concerned_organization specifies the Organization (see 4.2.294) that is affected by the result of the Activity.

EXAMPLE 22 – The production site that has to change the process due to a design change activity is an example for concerned_organization.

See 4.3.10 for the application assertion.

4.2.2.6 description

The description specifies additional information about the Activity.

See 4.3.9 for the application assertion.

The description need not be specified for a particular Activity.

If present, there shall be exactly one object that defines the description for an Activity.

4.2.2.7 id

The id specifies the identifier of the Activity.

4.2.2.8 planned_end_date

The planned_end_date specifies the date when the Activity is or was supposed to be finished.

See 4.3.7, 4.3.8, and 4.3.11 for the application assertions.

The planned_end_date need not be specified for a particular Activity.

If present, there shall be exactly one object that defines the planned_end_date for an Activity.

4.2.2.9 planned_start_date

The planned_start_date specifies the date when the Activity is or was supposed to be started.

See 4.3.7 and 4.3.8 for the application assertions.

The planned_start_date need not be specified for a particular Activity.

If present, there shall be exactly one object that defines the planned_start_date for an Activity.

4.2.2.10 requestor

The requestor specifies the Person (see 4.2.307) or Organization (see 4.2.294) that requested the Activity and the date the request was submitted.

See 4.3.6 for the application assertion.

The requestor need not be specified for a particular Activity.

4.2.2.11 resolved_request

The resolved_request specifies the Work_request (see 4.2.501) that is resolved by the Activity.

See 4.3.12 for the application assertion.

4.2.2.12 status

The status specifies the level of completion of the Activity.

EXAMPLE 23 – 'open' would be an example for status.

NOTE – The values and sequence of status are company specific.

The status need not be specified for a particular Activity.

4.2.2.13 supplying_organization

The supplying_organization specifies the Organization (see 4.2.294) that carries out the work.

See 4.3.10 for the application assertion.

4.2.3 Activity_element

An Activity_element is an item of work that is part of an Activity (see 4.2.2).

The data associated with an Activity_element are the following:

- activity;
- element;
- role.

4.2.3.1 activity

The activity specifies the Activity (see 4.2.2) the Activity_element belongs to.

See 4.3.13 for the application assertion.

4.2.3.2 element

The element specifies the piece of product data that is under work.

See 4.3.14, 4.3.15, 4.3.16, 4.3.17, 4.3.18, 4.3.19, 4.3.20, 4.3.21, 4.3.22, 4.3.23, 4.3.24, 4.3.25, 4.3.26, 4.3.27, 4.3.28, 4.3.29, 4.3.30, 4.3.31, 4.3.32, 4.3.33, 4.3.34, 4.3.35, 4.3.36, 4.3.37, 4.3.38, 4.3.39, 4.3.40, 4.3.41, 4.3.42, 4.3.43, 4.3.44, 4.3.45, 4.3.46, 4.3.47, and 4.3.48 for the application assertions.

There shall be exactly one object that defines the element for an Activity_element.

4.2.3.3 role

The role specifies the function that is performed by the Activity_element in the context of the concerned Activity. Where applicable the following values shall be used:

- 'input': The referenced element serves as initial data for the Activity;
- 'output': The referenced element is a result of the Activity.

4.2.4 Activity_method

An Activity_method is a procedure that may be used to solve a problem.

The data associated with an Activity_method are the following:

- consequence;
- description;
- name.

4.2.4.1 consequence

The consequence specifies the expected positive or negative effects of the application of a particular Activity_method.

The consequence need not be specified for a particular Activity_method.

4.2.4.2 description

The description specifies additional information that explicates the Activity_method in terms of nature of the Activity_method and/or in terms of means and methods to implement it.

See 4.3.49 for the application assertion.

There shall be exactly one object that defines the description for an Activity_method.

4.2.4.3 name

The name specifies the word or group of words by which the Activity_method is referred to.

See 4.3.49 for the application assertion.

There shall be exactly one object that defines the name for an Activity_method.

4.2.5 Activity_relationship

An Activity_relationship is a relationship between two Activity (see 4.2.2) objects.

The data associated with an Activity_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.5.1 description

The description specifies additional information about the Activity_relationship.

See 4.3.51 for the application assertion.

The description need not be specified for a particular Activity_relationship.

If present, there shall be exactly one object that defines the description for an Activity_-relationship.

4.2.5.2 related

The related specifies the second of the two Activity (see 4.2.2) objects related by an Activity_-relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.50 for the application assertion.

4.2.5.3 relating

The relating specifies the first of the two Activity (see 4.2.2) objects related by an Activity_-relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.50 for the application assertion.

4.2.5.4 relation_type

The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:

– 'decomposition': The application object defines a relationship where the related Activity (see 4.2.2) is one of the components into which the relating Activity is broken down;

– 'derivation': The application object defines a relationship where the related Activity is derived from the relating Activity;

EXAMPLE 24 – An Activity controlled by a Work_order with work order type 'manufacturing release' may be derived (is based on) from another Activity controlled by another Work_order with work order type 'design release'.

– 'sequence': The application object defines a relationship where the relating Activity shall be completed before the related Activity starts;

– 'precedence': The application object defines a relationship where the related Activity has higher priority than the relating Activity.

4.2.6 Actuator

An Actuator is the motivation of a Kinematic_pair (see 4.2.258) that causes its operation.

The data associated with an Actuator are the following:

- is_applied_to;
- name.

4.2.6.1 is_applied_to

The is_applied_to specifies the Kinematic_pair (see 4.2.258) that the Actuator is actuating.

See 4.3.52 for the application assertion.

4.2.6.2 name

The name specifies the word or group of words by which the Actuator is referred to.

See 4.3.53 for the application assertion.

There shall be exactly one object that defines the name for an Actuator.

4.2.7 Alias_identification

An Alias_identification is a mechanism to associate an object with an additional identifier that is used to identify the object of interest in a different context, either in another Organization (see 4.2.294), or in some other context. The scope of the Alias_identification shall be specified either by the attribute 'alias_scope' or by the attribute 'description'.

EXAMPLE 25 – An External_document (see 4.2.162) (e.g. a book) may have a unique document id (ISBN) and an Alias_identification as inventory number in the context of the inventory of a company.

The data associated with an Alias_identification are the following:

- alias_id;
- alias_scope;
- alias_version_id;
- description;
- is_applied_to.

4.2.7.1 alias_id

The alias_id specifies the identifier used in the context specified by the alias_scope, by the description, or by both.

4.2.7.2 alias_scope

The alias_scope specifies the Organization (see 4.2.294) in which the Alias_identification is valid.

See 4.3.62 for the application assertion.

The alias_scope need not be specified for a particular Alias_identification.

4.2.7.3 alias_version_id

The alias_version_id specifies the version of the object as known in the context of the Alias_identification. Alias_version_id only applies if 'version_id' is an attribute of the object of interest.

The alias_version_id need not be specified for a particular Alias_identification.

4.2.7.4 description

The description specifies the kind of the Alias_identification.

EXAMPLE 26 – The description may be e.g. 'inventory number'.

See 4.3.61 for the application assertion.

The description need not be specified for a particular Alias_identification.

If present, there shall be exactly one object that defines the description for an Alias_identification.

4.2.7.5 is_applied_to

The is_applied_to specifies the object that has an Alias_identification.

NOTE – The Item_version (see 4.2.248) is not included here. The alias of an Item_version is covered by Item_version_relationship (see 4.2.249).

See 4.3.54, 4.3.55, 4.3.56, 4.3.57, 4.3.58, 4.3.59, 4.3.60, 4.3.63, 4.3.64, 4.3.65, 4.3.66, 4.3.67, 4.3.68, 4.3.69, and 4.3.70 for the application assertions.

There shall be exactly one object that the Alias_identification is_applied_to.

4.2.8 Alternate_item_relationship

An Alternate_item_relationship is the relationship between two Item (see 4.2.236) objects specifying that all versions of the two related Item objects are interchangeable independent from their usage.

NOTE – Interchangeability usually refers to form, fit, function, and quality. Additional properties such as performance, noise, endurance, or reliability may also be considered as a prerequisite for the interchangeability. The two Item objects are interchangeable in both directions.

The data associated with an Alternate_item_relationship are the following:

- alternate;
- base;

– fulfilled_requirements.

4.2.8.1 alternate

The alternate specifies the Item (see 4.2.236) that may be used in place of the base Item.

See 4.3.71 for the application assertion.

4.2.8.2 base

The base specifies the Item (see 4.2.236) for which another Item may be used as an alternate.

See 4.3.71 for the application assertion.

4.2.8.3 fulfilled_requirements

The fulfilled_requirements specifies the word or group of words describing the requirements that are covered by both the base and the alternate and therefore are the basis for the statement of interchangeability.

See 4.3.72 for the application assertion.

There shall be exactly one object that defines the fulfilled_requirements for an Alternate_item_-relationship.

4.2.9 Angle_specification

An Angle_specification is an angle value and the specification of the element the angle is measured from.

EXAMPLE 27 – Figure 2 shows two examples of the use of an Angle_specification for the definition of a Form_feature_in_panel (see 4.2.188).

The data associated with an Angle_specification are the following:

– angle;

– angle_reference.

4.2.9.1 angle

The angle specifies the Feature_parameter (see 4.2.177) that represents the angle.

NOTE – Depending on the direction of the vector that serves as the reference for the angle measurement positive or negative values according to the definition of Angle_taper (see 4.2.10) should be used.

See 4.3.73 for the application assertion.

4.2.9.2 angle_reference

The angle_reference is the element the angle is measured from. The following values shall be used:

- 'punch direction': The angle is measured from the punch direction that is specified for the Process_feature_in_panel (see 4.2.334);
- 'profile normal': The angle is measured from a direction normal to the profile plane;
- 'surface normal': The angle is measured from the actual surface normals.

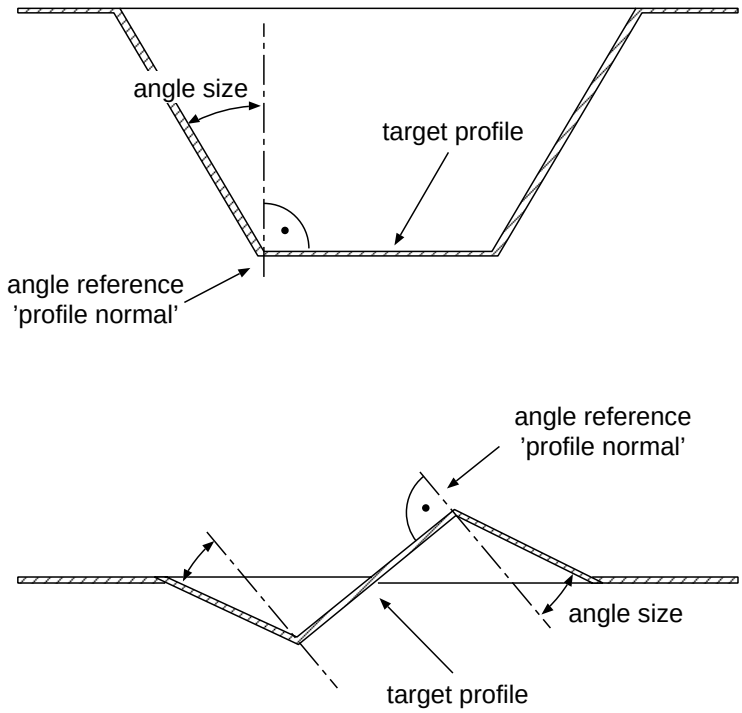


Figure 2 – Form features in panel defined by target boundaries and wall angles

4.2.10 Angle_taper

An Angle_taper is an angular deviation from a linear path. The Angle_taper is measured from the Linear_path (see 4.2.268) that is the sweep path of the feature the Angle_taper is applied to. The start of the Linear_path is the start of the taper.

EXAMPLE 28 – Figure 3 illustrates the application of an Angle_taper.

NOTE – Tapers are applied after general end conditions are applied.

The data associated with an Angle_taper are the following:

– angle.

The angle specifies the amount of grade from the start of an Angle_taper to the end of an Angle_taper. An angle between 0 and 90 degrees or between -90 and -180 degrees indicates that the profile grows larger along the sweep path. An angle between 0 and -90 degrees or between 90 and 180 degrees indicates that the profile grows smaller along the sweep path.

See 4.3.74 for the application assertion.

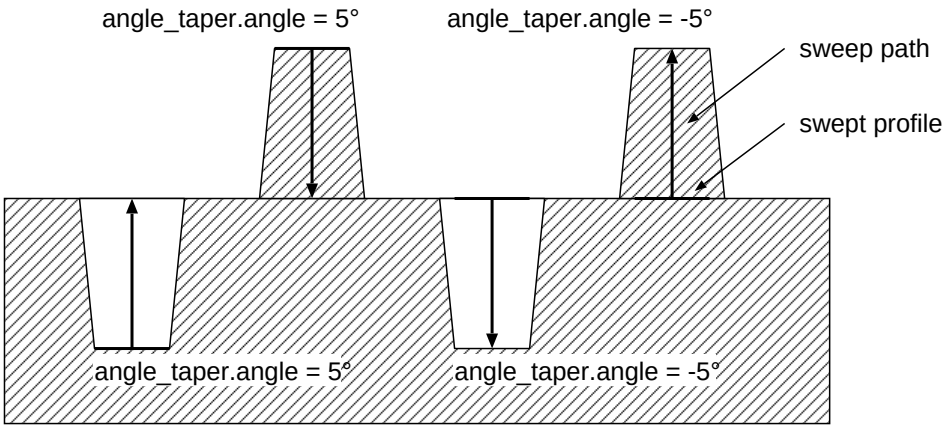


Figure 3 – Angle_taper

4.2.11 Angular_location_dimension

An Angular_location_dimension is the representation of an angular constraint concerning the position of two elements.

EXAMPLE 29 – Figure 4 illustrates the application of the attributes of an Angular_location_dimension.

An Angular_location_dimension is a type of Location_dimension (see 4.2.270).

The data associated with an Angular_location_dimension are the following:

– major_angle;

– mirror_target;

– origin_determination.

4.2.11.1 major_angle

The major_angle specifies a Boolean value that indicates the way the dimension is defined. A value of 'true' specifies the angle to be the larger of the two angles formed by the two elements that are referred to by the Angular_location_dimension. This attribute is used in combination with the other attributes to determine the angle that is constrained.

4.2.11.2 mirror_target

The mirror_target specifies a Boolean value that indicates whether the element that defines the target of the limitation is to be mirrored in order to define the angle whose dimension is specified. A value of 'true' indicates that the target object is mirrored about a plane orthogonal to it.

4.2.11.3 origin_determination

The origin_determination specifies the method used to identify the first of the two elements that form the angle. The following values shall be used:

- 'unchanged': The Angular_location_dimension is measured from the origin as identified;

- 'mirror': The Angular_location_dimension is measured from the origin that is mirrored about a plane orthogonal to it through the intersection point of origin and target;

- 'normal': The Angular_location_dimension is measured from a line or plane that is normal to the identified origin through the intersection point of origin and target.

4.2.12 Angular_size_dimension

An Angular_size_dimension is a dimension representing an angular size of an aspect of a part.

EXAMPLE 30 – Figure 5 shows the difference between an Angular_size_dimension and an Angular_location_dimension (see 4.2.11).

An Angular_size_dimension is a type of Size_dimension (see 4.2.405).

The data associated with an Angular_size_dimension are the following:

- full;

- major_angle.

4.2.12.1 full

The full specifies the method used to establish the Angular_size_dimension. A value of 'true' specifies the angle is established between the two sides of the angular element. A value of 'false' specifies the angle is established between the bisector line and the angular element.

4.2.12.2 major_angle

The major_angle specifies which one of the two possible angles is meant. A value of 'true' specifies the angle to be the larger of the two angles formed by the angular element.

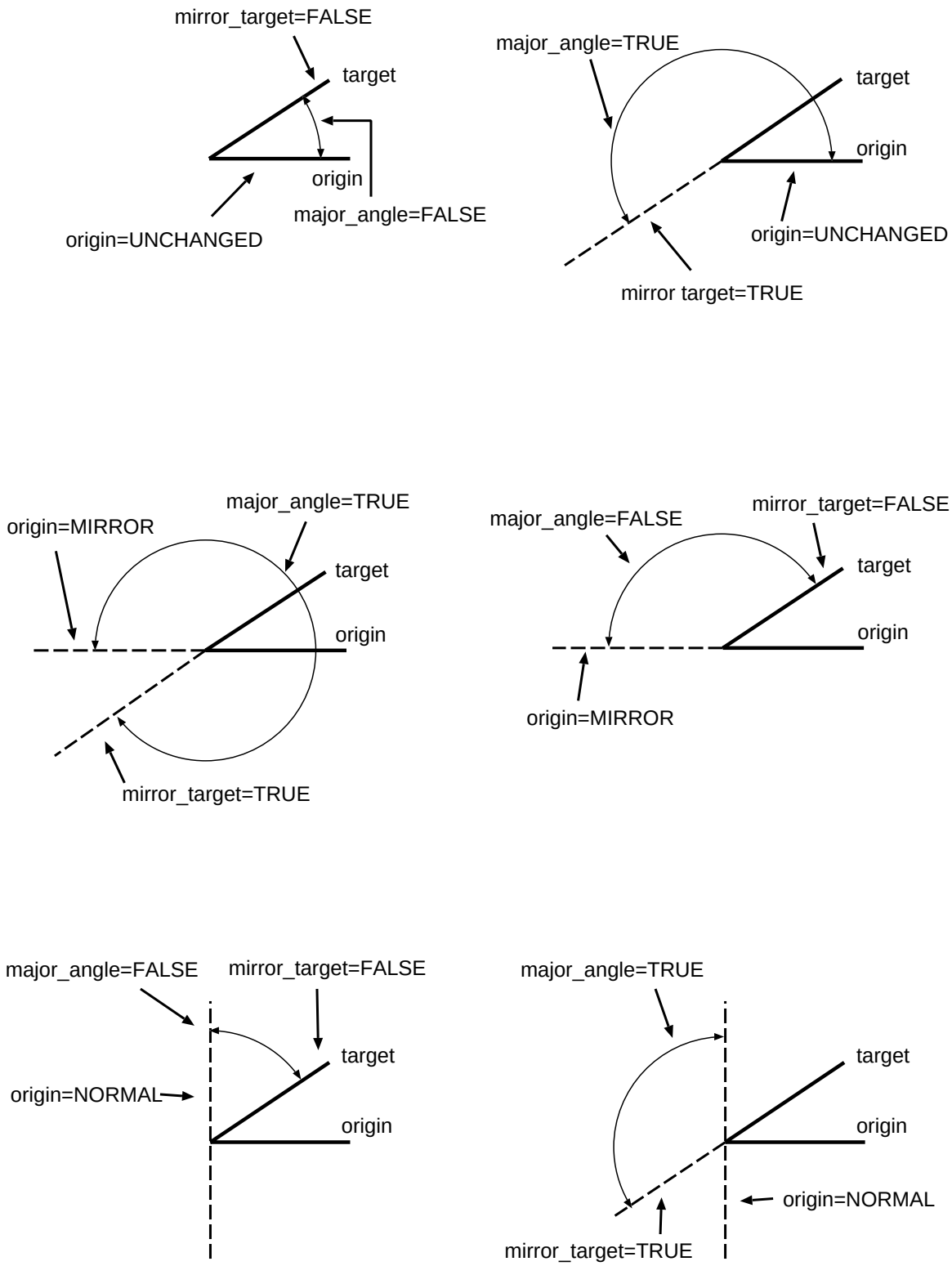


Figure 4 – Angular_location_dimension

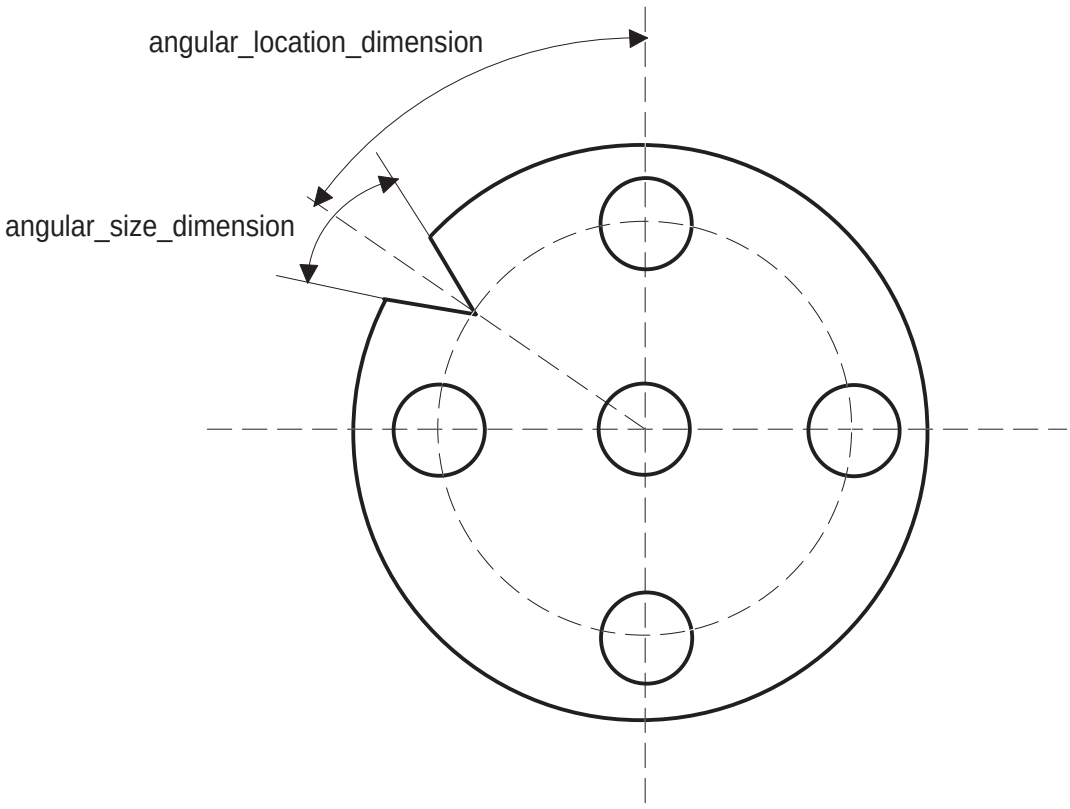


Figure 5 – Angular_location_dimension versus angular_size_dimension

4.2.13 Angularity_tolerance

An Angularity_tolerance is a tolerance for the condition of a surface or axis at a specified angle (other than 90 degrees) from a datum plane or axis. An Angularity_tolerance specifies one of the following:

- a tolerance zone defined by two parallel planes at the specified basic angle from a datum plane or axis within which the surface of the considered feature must lie;

- a tolerance zone defined by two parallel planes at the specified basic angle from a datum plane or axis within which the axis of the considered feature must lie.

An Angularity_tolerance is a type of Geometric_tolerance (see 4.2.216).

The data associated with an Angularity_tolerance are the following:

- datum;

- segment_size.

4.2.13.1 datum

The datum specifies the Datum_feature (see 4.2.117) or the Compound_datum (see 4.2.80) that defines the reference for this type of geometric tolerance.

See 4.3.75 for the application assertion.

4.2.13.2 segment_size

The segment_size specifies the length (or the edge length) of a segment if the geometric tolerance is not applied to the total length/area of an axis/surface but on a unit base (e.g. to each segment with a length of 20 mm).

The segment_size need not be specified for a particular Angularity_tolerance.

4.2.14 Annotation_curve

An Annotation_curve is a two-dimensional trimmed curve that is used to annotate a drawing or a shape model.

An Annotation_curve is a type of Annotation_element (see 4.2.15).

Each Annotation_curve may be one of the following: a Projection_curve (see 4.2.359), a Dimension_curve (see 4.2.142), or a Leader (see 4.2.262).

The data associated with an Annotation_curve are the following:

- annotation_curve_type.

The annotation_curve_type specifies the kind of annotation curve related to its usage.

EXAMPLE 31 – 'centre line' is an example for annotation_curve_type.

The annotation_curve_type need not be specified for a particular Annotation_curve.

4.2.15 Annotation_element

An Annotation_element is Styled_element (see 4.2.427) that represents annotation.

An Annotation_element is a type of Styled_element (see 4.2.427).

Each Annotation_element is either a Text (see 4.2.454), a Dimension (see 4.2.140), an Annotation_subfigure (see 4.2.17), an Annotation_curve (see 4.2.14), an Annotation_symbol (see 4.2.18), a Draughting_callout (see 4.2.148), or a Fill_area (see 4.2.180).

4.2.16 Annotation_model

An Annotation_model is a set of annotation elements.

An Annotation_model is a type of Styled_model (see 4.2.429).

The data associated with an Annotation_model are the following:

- is_defined_in.

The is_defined_in specifies the Cartesian_coordinate_space (see 4.2.44) the Annotation_model is defined in.

NOTE – An Annotation_model is defined in its own coordinate space since it does not refer to a Geometric_model (see 4.2.211) that has a coordinate space that could be shared.

See 4.3.76 for the application assertion.

4.2.17 Annotation_subfigure

An Annotation_subfigure is the occurrence of a template that contains annotation elements. An Annotation_subfigure is located within the coordinate system of a Drawing_sheet (see 4.2.152), a Styled_model (see 4.2.429), or another Annotation_subfigure, View_area (see 4.2.491), or View (see 4.2.490).

EXAMPLE 32 – Figure 6 depicts the Annotation_subfigure.

An Annotation_subfigure is a type of Annotation_element (see 4.2.15).

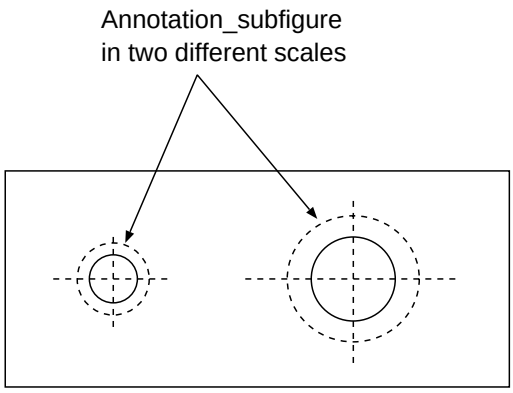


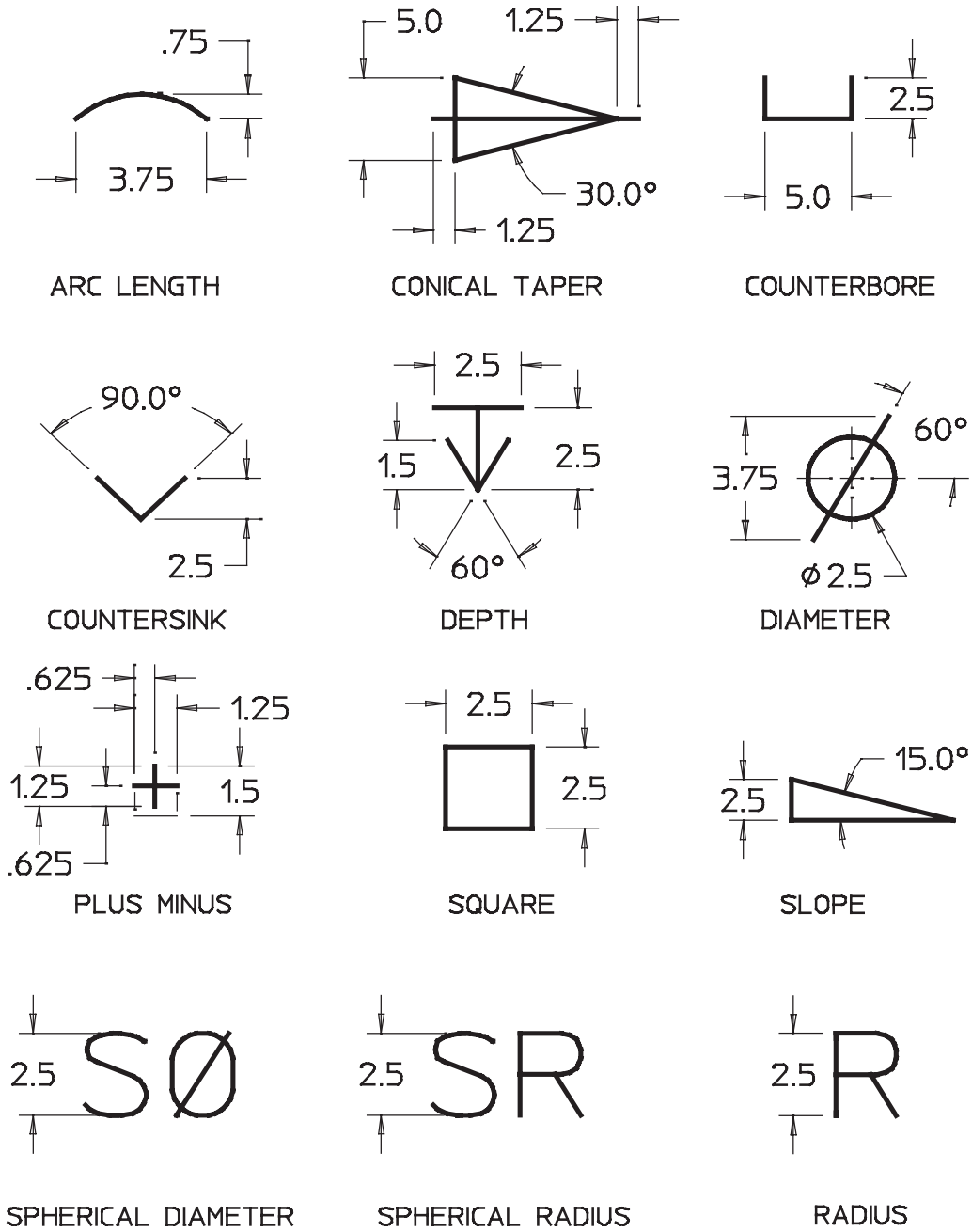
Figure 6 – Annotation_subfigure for a hole on a technical drawing

4.2.18 Annotation_symbol

An Annotation_symbol is a combination of annotation elements that, as collection, represents a recognized annotation concept. Annotation_symbol objects are dimension symbols, geometrical tolerance symbols, surface condition symbols, or terminator symbols and they are either user defined, externally defined, or predefined. The set of predefined symbols comprises the symbols for geometrical tolerances as defined in ISO 7083 as well as the dimension symbols shown in figure 7 and the terminator symbols shown in figure 8.

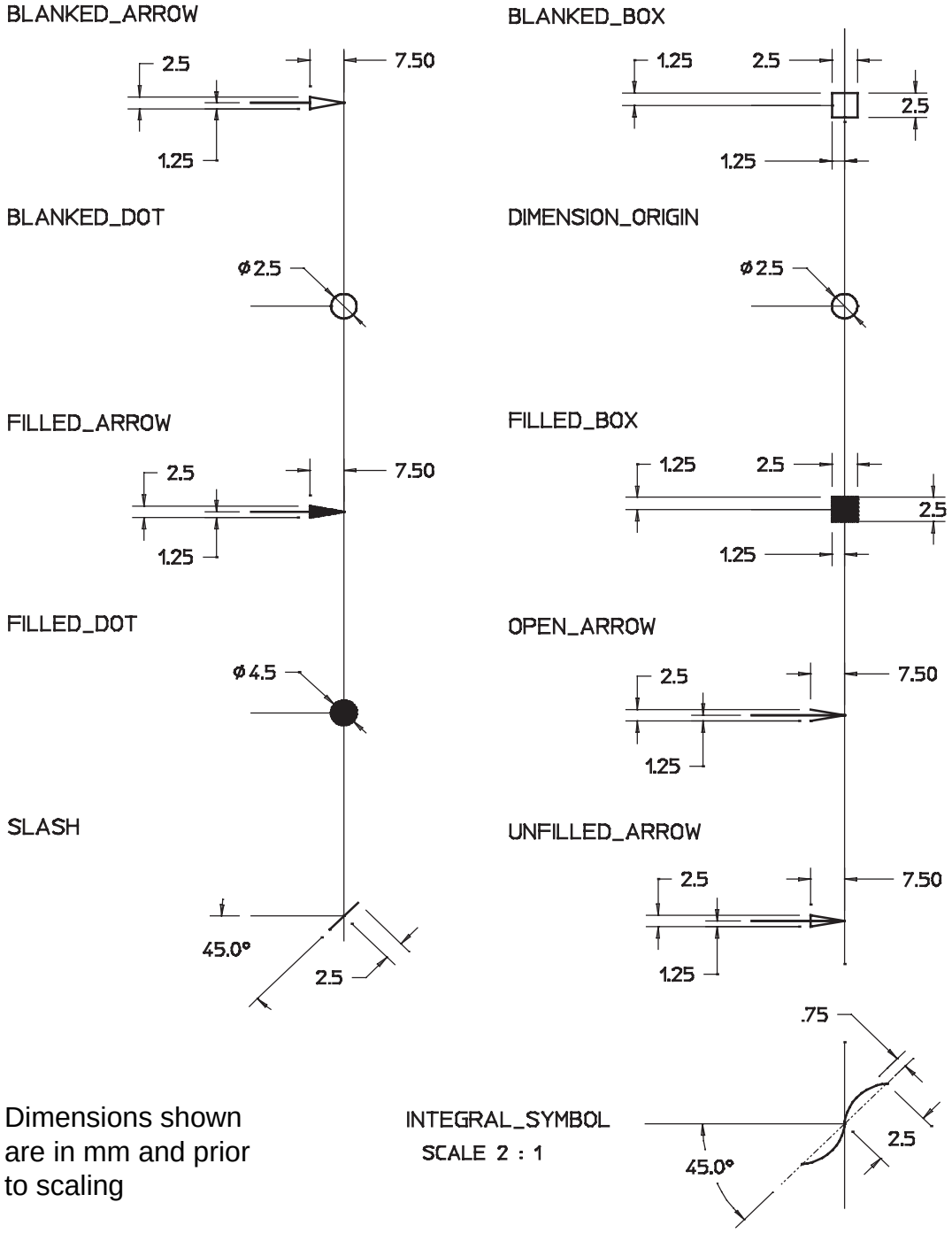
An Annotation_symbol is a type of Annotation_element (see 4.2.15).

Each Annotation_symbol may be an Associated_annotation_symbol (see 4.2.29).



(Dimensions shown are in mm and prior to scaling)

Figure 7 – Predefined dimension symbols



Dimensions shown are in mm and prior to scaling

Figure 8 – Predefined terminator symbols

4.2.19 Appearance_assignment

An Appearance_assignment governs the visual appearance of geometric elements and annotation. The appearance can be defined explicitly for each represented element or for a context in which the elements are represented.

The different kinds of the context dependent appearance are Model_image (see 4.2.284), View (see 4.2.490), View_area (see 4.2.491), Drawing_sheet (see 4.2.152), Group (see 4.2.222) and Layer (see 4.2.261). Several appearance components for the representation of the different kinds of elements can be associated with the Appearance_assignment.

Each Appearance_assignment may be a Context_dependent_appearance_assignment (see 4.2.98).

The data associated with an Appearance_assignment are the following:

- component.

The component specifies the various appearance components to define the appearance of geometric and annotation objects.

See 4.3.77, 4.3.78, 4.3.79, 4.3.80, 4.3.81, 4.3.82, 4.3.83, and 4.3.84 for the application assertions.

There shall be at least one object that defines the component for an Appearance_assignment.

4.2.20 Application_context

An Application_context is a shared universe of discourse.

The data associated with an Application_context are the following:

- application_domain;
- description;
- life_cycle_stage.

4.2.20.1 application_domain

The application_domain is the identification of the applications for which an object may be relevant.

EXAMPLE 33 – The application_domain may be e.g., 'electrical design' or 'engineering analysis'.

4.2.20.2 description

The description specifies additional information about the Application_context.

See 4.3.85 for the application assertion.

The description need not be specified for a particular Application_context.

If present, there shall be exactly one object that defines the description for an Application_context.

4.2.20.3 life_cycle_stage

The life_cycle_stage is the specification of the general stage in the product life cycle to which the concerned items belong.

EXAMPLE 34 – 'The life_cycle_stage may be e.g., 'preliminary design', or 'pre-series'.

4.2.21 Approval

An Approval is a judgement concerning the quality of those product data that are subject of the Approval. An Approval represents a statement made by technical personnel or management personnel if certain requirements are met.

The data associated with an Approval are the following:

- actual_date;
- approval_status;
- description;
- is_applied_to;
- is_approved_by;
- nominal_date.

4.2.21.1 actual_date

The actual_date specifies the date when the Approval was actually performed. If this attribute is absent, the approval has not yet occurred, i.e. it is pending.

See 4.3.98 for the application assertion.

The actual_date need not be specified for a particular Approval.

4.2.21.2 approval_status

The approval_status specifies the judgement made about the product data that is the subject of this Approval.

See 4.3.89 for the application assertion.

4.2.21.3 description

The description specifies additional information about the Approval.

See 4.3.111 for the application assertion.

The description need not be specified for a particular Approval.

If present, there shall be exactly one object that defines the description for an Approval.

4.2.21.4 is_applied_to

The is_applied_to specifies the object to which the Approval is assigned.

See 4.3.86, 4.3.87, 4.3.88, 4.3.90, 4.3.91, 4.3.92, 4.3.93, 4.3.94, 4.3.95, 4.3.96, 4.3.99, 4.3.100, 4.3.101, 4.3.102, 4.3.103, 4.3.104, 4.3.105, 4.3.106, 4.3.107, 4.3.108, 4.3.109, 4.3.110, 4.3.112, 4.3.113, 4.3.114, 4.3.115, 4.3.116, 4.3.117, 4.3.118, 4.3.119, 4.3.120, 4.3.121, 4.3.122, 4.3.123, 4.3.124, 4.3.125, and 4.3.126 for the application assertions.

There shall be at least one object that the Approval is_applied_to.

4.2.21.5 is_approved_by

The is_approved_by specifies the person or organization responsible for an Approval and the dates of the Approval.

See 4.3.97 for the application assertion.

4.2.21.6 nominal_date

The nominal_date specifies the date when the Approval is or was supposed to be performed.

See 4.3.98 for the application assertion.

The nominal_date need not be specified for a particular Approval.

4.2.22 Approval_relationship

An Approval_relationship is a relationship between two Approval (see 4.2.21) objects.

The data associated with an Approval_relationship are the following:

- related;
- relating;
- relation_type.

4.2.22.1 related

The related specifies the second of the two Approval (see 4.2.21) objects related by the Approval_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.127 for the application assertion.

4.2.22.2 relating

The relating specifies the first of the two Approval (see 4.2.21) objects related by the Approval_-relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.127 for the application assertion.

4.2.22.3 relation_type

The relation_type specifies the meaning of the relationship.

4.2.23 Approval_status

An Approval_status is the state of acceptance of some product data.

NOTE – The values of Approval_status are company specific.

The data associated with an Approval_status are the following:

- name;
- used_classification_system.

4.2.23.1 name

The name specifies the word or abbreviation which is used to refer to the Approval_status.

4.2.23.2 used_classification_system

The used_classification_system specifies the Classification_system (see 4.2.70) that contains the information about how to interpret the Approval_status.

See 4.3.128 for the application assertion.

The used_classification_system need not be specified for a particular Approval_status.

4.2.24 Archived_document

An Archived_document is a piece of product data that is archived in a non-digital form.

EXAMPLE 35 – Paper plots of technical drawings, microfiche, or paper documents such as calculations or test reports are examples for an Archived_document.

An Archived_document is a type of External_document (see 4.2.162).

The data associated with an Archived_document are the following:

- archiving_format;
- found_in.

4.2.24.1 archiving_format

The archiving_format specifies the storage medium.

EXAMPLE 36 – 'paper plot A0' and 'microfiche' are examples for the archiving_format.

4.2.24.2 found_in

The found_in specifies the information necessary to access the Archived_document.

EXAMPLE 37 – In case of a paper plot, this information may be the organizational address of the cabinet where the plot is stored. It may also be the name and address of the responsible person.

4.2.25 Area_placed_annotation

An Area_placed_annotation is a two-dimensional annotation element that is defined in the two-dimensional coordinate space of a View_area (see 4.2.491) or Drawing_sheet (see 4.2.152).

EXAMPLE 38 – The drawing title box or drawing frame are examples for Area_placed_annotation objects on a drawing sheet.

The data associated with an Area_placed_annotation are the following:

- appearance;
- element;
- invisible;
- reference_area.

4.2.25.1 appearance

The appearance specifies the presentation characteristics for the Annotation_element (see 4.2.15) that is placed.

See 4.3.130 for the application assertion.

4.2.25.2 element

The element specifies the presented Annotation_element (see 4.2.15).

See 4.3.129 for the application assertion.

4.2.25.3 invisible

The invisible specifies if the Area_placed_annotation is invisible.

4.2.25.4 reference_area

The reference_area specifies the View_area (see 4.2.491) or Drawing_sheet (see 4.2.152) where the Annotation_element (see 4.2.15) is located.

See 4.3.131 and 4.3.132 for the application assertions.

There shall be exactly one object that defines the reference_area for an Area_placed_annotation.

4.2.26 Assembly_component_relationship

An Assembly_component_relationship is the relation between an Assembly_definition (see 4.2.27) and an Item_instance (see 4.2.241) representing a constituent of the assembly.

NOTE – The constituent may also be an assembly.

An Assembly_component_relationship is a type of Item_definition_instance_relationship (see 4.2.239).

Each Assembly_component_relationship may be one of the following: a Next_higher_assembly (see 4.2.287), a Promissory_usage (see 4.2.360), or a Specified_higher_usage (see 4.2.419).

The data associated with an Assembly_component_relationship are the following:

- relating;
- transformation.

4.2.26.1 relating

The relating specifies the Assembly_definition (see 4.2.27) that has subordinate constituents.

See 4.3.133 for the application assertion.

4.2.26.2 transformation

The transformation specifies the Geometric_model_relationship_with_transformation (see 4.2.213) or the Template_instance (see 4.2.453) that has the Transformation (see 4.2.472) information which is used to locate and orient the constituent in the coordinate space of the Assembly_definition (see 4.2.27). In the case of a Template_instance (see 4.2.453) the scale factor shall be omitted or set to 1.0.

See 4.3.134 and 4.3.135 for the application assertions.

The transformation need not be specified for a particular Assembly_component_relationship.

If present, there shall be exactly one object that defines the transformation for an Assembly_component_relationship.

4.2.27 Assembly_definition

An Assembly_definition is a definition of an Item_version (see 4.2.248) that contains other subordinate Item_version objects.

An Assembly_definition is a type of Design_discipline_item_definition (see 4.2.134).

The data associated with an Assembly_definition are the following:

- assembly_type.

The assembly_type specifies the kind of the Assembly_definition.

EXAMPLE 39 – 'functional assembly','manufacturing assembly','design assembly' are examples of an assembly_type.

4.2.28 Assembly_substitute_relationship

An Assembly_substitute_relationship is a relationship that indicates that one Assembly_component_relationship (see 4.2.26) may substitute another Assembly_component_relationship. The subject of the substitution are the related item instance objects of both Assembly_component_relationship (see 4.2.26) objects. The relating Assembly_definition (see 4.2.27) shall be the same in both Assembly_substitute_relationship objects.

The data associated with an Assembly_substitute_relationship are the following:

- base;
- description;
- substitute.

4.2.28.1 base

The base specifies the Assembly_component_relationship (see 4.2.26) that is replaceable.

See 4.3.136 for the application assertion.

4.2.28.2 description

The description specifies additional information about the Assembly_substitute_relationship.

See 4.3.137 for the application assertion.

The description need not be specified for a particular Assembly_substitute_relationship.

If present, there shall be exactly one object that defines the description for an Assembly_substitute_relationship.

4.2.28.3 substitute

The substitute specifies the Assembly_component_relationship (see 4.2.26) that may be used instead of the base Assembly_component_relationship.

See 4.3.136 for the application assertion.

4.2.29 Associated_annotation_symbol

An Associated_annotation_symbol is an Annotation_symbol (see 4.2.18) that is associated to a geometric element, an annotation curve, a fill area, a text, an annotation subfigure, or another annotation symbol.

An Associated_annotation_symbol is a type of Annotation_symbol (see 4.2.18).

Each Associated_annotation_symbol may be one of the following: a Geometric_tolerance_symbol (see 4.2.218) or a Surface_condition_symbol (see 4.2.438).

The data associated with an Associated_annotation_symbol are the following:

- is_associated_to.

The is_associated_to specifies the element where the Associated_annotation_symbol is associated to.

See 4.3.138, 4.3.139, 4.3.140, 4.3.141, 4.3.142, and 4.3.143 for the application assertions.

4.2.30 Associated_fill_area

An Associated_fill_area is a bounded area containing colouring, hatching, or tiling that indicate its extent and content. The boundary of the Associated_fill_area is defined by closed annotation curves. The annotation curves defining the boundary of the Associated_fill_area are derived from geometric elements of the shape model or other annotation curves.

An Associated_fill_area is a type of Fill_area (see 4.2.180).

The data associated with an Associated_fill_area are the following:

- is_derived_from.

The is_derived_from specifies the geometric elements or annotation curves from which the boundary curves of the Associated_fill_area are derived.

See 4.3.144 and 4.3.145 for the application assertions.

There shall be at least one object that the Associated_fill_area is_derived_from.

4.2.31 Associated_leader

An Associated_leader is a leader that is associated with a Detailed_geometric_model_element (see 4.2.136) or with another Annotation_element (see 4.2.15).

An Associated_leader is a type of Leader (see 4.2.262).

The data associated with an Associated_leader are the following:

- target.

The target specifies the associated element.

See 4.3.146, 4.3.147, 4.3.148, 4.3.149, 4.3.150, and 4.3.151 for the application assertions.

There shall be exactly one object that defines the target for an Associated_leader.

4.2.32 Associated_projection_curve

An Associated_projection_curve is a projection curve that is associated with a Detailed_geometric_-model_element (see 4.2.136) or with another Annotation_element (see 4.2.15).

An Associated_projection_curve is a type of Projection_curve (see 4.2.359).

The data associated with an Associated_projection_curve are the following:

- is_projected_to.

The is_projected_to specifies the associated element.

See 4.3.152, 4.3.153, 4.3.154, 4.3.155, 4.3.156, and 4.3.157 for the application assertions.

There shall be exactly one object that the Associated_projection_curve is_projected_to.

4.2.33 Axis_placement

An Axis_placement is the representation of three orthogonal axes defined in a reference coordinate system.

4.2.34 B_rep

A B_rep is defined by exactly one outer Closed_shell (see 4.2.73) and zero or more inner Closed_-shell objects.

A B_rep is a type of Detailed_geometric_model_element (see 4.2.136).

The data associated with a B_rep are the following:

- outer;
- void.

4.2.34.1 outer

The outer specifies a Closed_shell (see 4.2.73) defining the exterior boundary of the B_rep.

See 4.3.158 for the application assertion.

4.2.34.2 void

The void specifies a Closed_shell (see 4.2.73) that represents an internal cavity of the B_rep.

NOTE – This shell has a normal pointing into the empty space inside.

See 4.3.158 for the application assertion.

4.2.35 B_rep_model

A B_rep_model is a collection of B_rep (see 4.2.34) objects.

A B_rep_model is a type of Geometric_model (see 4.2.211).

The data associated with a B_rep_model are the following:

– element.

The element specifies a set of B_rep (see 4.2.34) objects that form the B_rep_model.

See 4.3.159 for the application assertion.

4.2.36 B_spline_curve

A B_spline_curve is a general polynomial or a rational parametric sculptured curve of various degrees and segments, defined in terms of control points, associated basic functions, weights, and knots. Representing a uniform polynomial parametric curve a B_spline_curve is defined by a list of control points and the associated basis functions. The knot values can be derived and the weights are equal. To represent a general rational parametric curve, the B_spline_curve is defined by a list of control points and the associated basis functions as well, but additionally weights and multiplicities of knots have to be defined. In this case the B_spline_curve describes a NURBS-curve, i.e., Non Uniform Rational B-Spline. In case of equal weights and a non-uniform knot vector, the B_spline_curve would represent a non uniform polynomial parametric curve.

EXAMPLE 40 – A circle is an example for a general rational parametric curve.

A B_spline_curve is a type of Curve (see 4.2.102).

4.2.37 B_spline_surface

A B_spline_surface is a bounded surface that is used to represent a wide variety of sculptured surfaces. With appropriate attribute values it is capable of representing single span or spline surfaces of explicit polynomial, rational Bézier, or B-spline types, such as NURBS. A B_spline_surface is defined in terms of control points, associated basis functions, weights, and knots. A B_spline_surface shall not self-intersect.

A B_spline_surface is a type of Surface (see 4.2.434).

4.2.38 Bead_end_type

A Bead_end_type is the definition of a kind of end that characterizes a Bead_feature (see 4.2.39).

Each Bead_end_type is either a Linear_bead_end_type (see 4.2.266), an Open_bead_end_type (see 4.2.291), or a Radiused_bead_end_type (see 4.2.366).

The data associated with a Bead_end_type are the following:

– is_applied_to.

The is_applied_to specifies to which extremity of the path the Bead_end_type applies. The following values shall be used:

– 'start': The Bead_end_type is applied to the extremity of the Bead_feature (see 4.2.39) that corresponds to the start of the defining Path (see 4.2.304);

– 'end': The Bead_end_type is applied to the extremity of the Bead_feature that corresponds to the end of the defining Path.

4.2.39 Bead_feature

A Bead_feature is a feature that is applied to panel parts to increase the stiffness of the part.

EXAMPLE 41 – Figure 9 shows a Bead_feature and illustrates the results of the application of various Bead_end_type (see 4.2.38) conditions.

A Bead_feature is a type of Process_feature_in_panel (see 4.2.334).

The data associated with a Bead_feature are the following:

– base_radius;

– bead_profile;

– end_condition;

– maximum_depth;

– profile_placement;

– sweep_path.

4.2.39.1 base_radius

The base_radius specifies the blend radius where the Bead_feature meets the panel surface.

See 4.3.161 for the application assertion.

The base_radius need not be specified for a particular Bead_feature.

4.2.39.2 bead_profile

The bead_profile specifies the profile that determines the cross section of the Bead_feature.

See 4.3.162 for the application assertion.

4.2.39.3 end_condition

The end_condition specifies the Bead_end_type (see 4.2.38) objects that define the types of the bead's ends. End condition applies to both ends of the Bead_feature.

See 4.3.160 for the application assertion.

4.2.39.4 maximum_depth

The maximum_depth specifies the distance in the punch direction from the path to the farthest point of contact at the bottom of the Bead_feature.

The maximum_depth need not be specified for a particular Bead_feature.

4.2.39.5 profile_placement

The profile_placement specifies a geometric Transformation (see 4.2.472) that may be used to place the defining Open_profile (see 4.2.292) in the Bead_feature's coordinate space.

See 4.3.164 for the application assertion.

The profile_placement need not be specified for a particular Bead_feature.

4.2.39.6 sweep_path

The sweep_path specifies the course along which the bead_profile is swept.

See 4.3.163 for the application assertion.

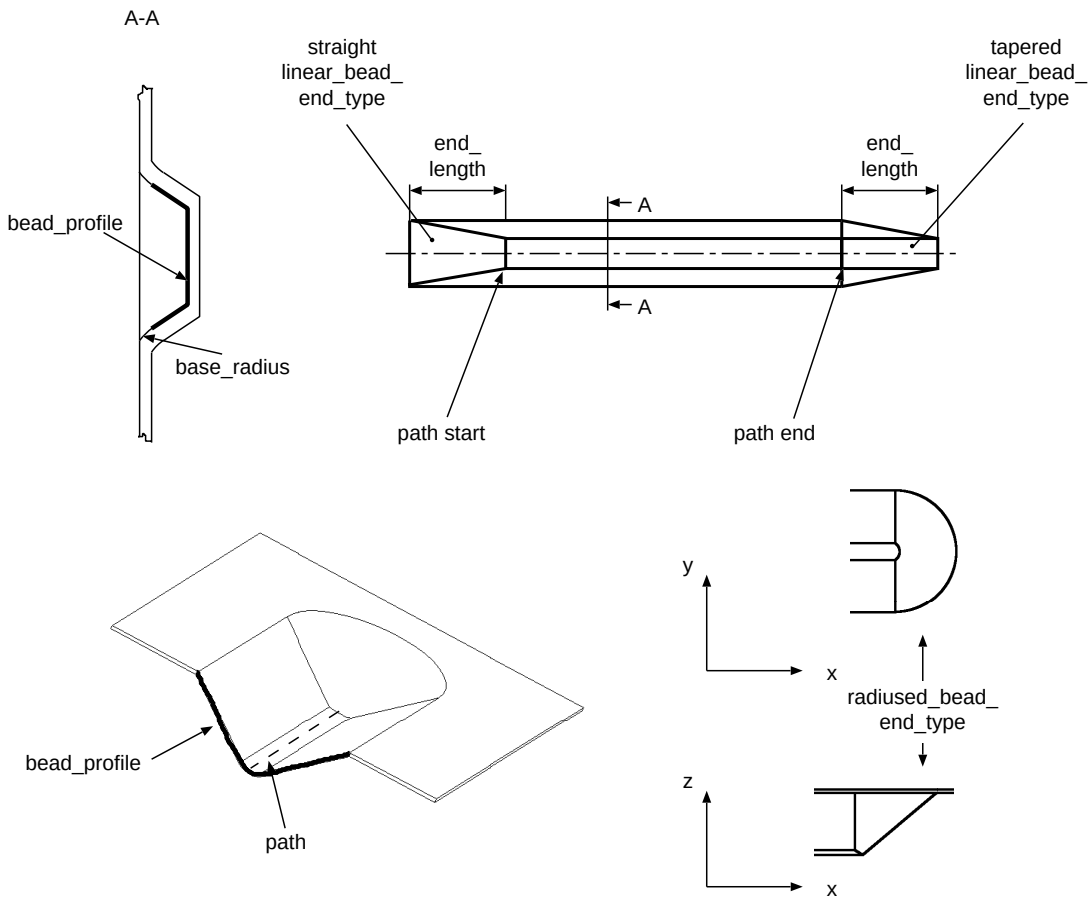


Figure 9 – Bead_feature

4.2.40 Block

A Block is a solid rectangular parallelepiped defined by a location and placement coordinate system. The Block is specified by the positive lengths x, y, and z along the axes of the placement coordinate system, and has one vertex at the origin of the placement coordinate system.

A Block is a type of Csg_primitive (see 4.2.101).

4.2.41 Boolean_result

A Boolean_result is the result of a regularised operation on two solids to create a new solid.

NOTE – A Boolean_result may create a new solid with more than one solid body.

A Boolean_result is a type of Detailed_geometric_model_element (see 4.2.136).

The data associated with a Boolean_result are the following:

- boolean_operation;
- first_operand;
- second_operand.

4.2.41.1 boolean_operation

The boolean_operation specifies the type of operation that is applied on the two solids. The operation types permitted are: The following values shall be used:

- 'union': creates a new solid that contains all the points that are either in the first operand or in the second operand or both;
- 'intersection': creates a new solid that is the regularisation of the set of all points that are in both the first operand and the second operand;
- 'difference': creates a new solid that is the regularisation of the set of all points, which are in the first operand but not in the second operand.

4.2.41.2 first_operand

The first_operand specifies the the first solid for the Boolean operation.

See 4.3.165, 4.3.166, 4.3.167, 4.3.168, 4.3.169, 4.3.170, and 4.3.171 for the application assertions.

There shall be exactly one object that defines the first_operand for a Boolean_result.

4.2.41.3 second_operand

The second_operand specifies the second solid for the Boolean operation.

See 4.3.165, 4.3.166, 4.3.167, 4.3.168, 4.3.169, 4.3.170, and 4.3.171 for the application assertions.

There shall be exactly one object that defines the second_operand for a Boolean_result.

4.2.42 Boss_feature

A Boss_feature is an enclosed shape that protrudes from a part's surface in the direction of the sweep path. The intersection of the Boss_feature and the part's surface may but need not have radius shaped blend between them.

A Boss_feature is a type of Process_feature_in_solid (see 4.2.335).

Each Boss_feature is either a Circular_boss (see 4.2.56), a Rectangular_boss (see 4.2.368), or a General_boss (see 4.2.190).

The data associated with a Boss_feature are the following:

- fillet_radius;
- profile_placement;
- sweep_path;
- taper;
- top_condition.

4.2.42.1 fillet_radius

The fillet_radius specifies a radius shape blend between a Boss_feature and the surrounding surface at its base. The fillet is applied after the Boss_feature (see 4.2.42) itself is placed on the part and after a potential taper is applied.

For the creation of a fillet the following rules apply:

If R_f is the fillet radius to be created at the bottom of the boss and C the path along which the centre of a sphere S defining the fillet is moving then

- the fillet is created by rolling the sphere S with radius R_f along the boss with a constant contact on the surface on which the boss is placed;
- C shall be such that in any point of C the area of contact between S and the boss is either a point or an arc wise connected set of points.

See 4.3.176 for the application assertion.

The fillet_radius need not be specified for a particular Boss_feature.

4.2.42.2 profile_placement

The profile_placement specifies a geometric Transformation (see 4.2.472) that may be used to place the defining Closed_profile (see 4.2.71) in the Boss_feature's coordinate space.

See 4.3.178 for the application assertion.

The profile_placement need not be specified for a particular Boss_feature.

4.2.42.3 sweep_path

The sweep_path specifies the Linear_path (see 4.2.268) that is used to sweep the Closed_profile (see 4.2.71).

See 4.3.177 for the application assertion.

4.2.42.4 taper

The taper specifies a variation of the size of the Boss_feature. If present the variation shall be specified either by an Angle_taper (see 4.2.10) or by a Directed_taper (see 4.2.145) or in the case of a Circular_boss (see 4.2.56) by a Diameter_taper (see 4.2.138). If no taper is specified the cross section at the start of the sweep path and at the end of the sweep path shall be of identical shape.

See 4.3.172, 4.3.174, and 4.3.175 for the application assertions.

The taper need not be specified for a particular Boss_feature.

If present, there shall be exactly one object that defines the taper for a Boss_feature.

4.2.42.5 top_condition

The top_condition specifies the shape at the top of the Boss_feature (see 4.2.42).

See 4.3.173 for the application assertion.

4.2.43 Boss_top_condition

A Boss_top_condition is the definition of a kind of top that characterizes a Boss_feature (see 4.2.42).

Each Boss_top_condition is either a Planar_top_condition (see 4.2.316) or a General_top_condition (see 4.2.209).

The data associated with a Boss_top_condition are the following:

- is_applied_to.

The is_applied_to specifies to which end of the path the Boss_top_condition applies. The following values shall be used:

- 'start': The Boss_top_condition is applied to that extremity of the Boss_feature (see 4.2.42) that corresponds to the start of the defining Linear_path (see 4.2.268);

- 'end': The Boss_top_condition is applied to that extremity of the Boss_feature that corresponds to the end of the defining Linear_path.

4.2.44 Cartesian_coordinate_space

A Cartesian_coordinate_space is the coordinate space where geometric and annotation elements are defined in. It is either two-dimensional or three-dimensional. An origin for coordinate values is implicitly defined. Cartesian Transformation (see 4.2.472) objects are used to specify the relationships between coordinate spaces. The units applicable to the coordinate values of elements defined in the Cartesian_coordinate_space are specified.

Each Cartesian_coordinate_space is either a Cartesian_coordinate_space_3d (see 4.2.46) or a Cartesian_coordinate_space_2d (see 4.2.45).

The data associated with a Cartesian_coordinate_space are the following:

- unit.

The unit specifies the various kinds of unit in that any values are measured. In the case where geometric elements are defined in the Cartesian_coordinate_space there shall be at least two units specified, the length unit and the plane angle unit. The same length unit is applied to each coordinate direction. Only one unit of a kind shall be specified.

NOTE – If elements with different units are required they have to be separated into different models with their own Cartesian_coordinate_space.

EXAMPLE 42 – A length measure unit measured in inches and an angle measure unit measured in degrees are examples for two attributes 'unit' assigned to the same Cartesian_coordinate_space.

There may be more than two unit attributes for a Cartesian_coordinate_space.

4.2.45 Cartesian_coordinate_space_2d

A Cartesian_coordinate_space_2d is a coordinate space defined in two-dimensional space only.

NOTE – Each two-dimensional geometric and annotation element is defined in a Cartesian_coordinate_space_2d.

EXAMPLE 43 – The annotation elements associated in a View_area (see 4.2.491) are defined in a Cartesian_coordinate_space_2d.

A Cartesian_coordinate_space_2d is a type of Cartesian_coordinate_space (see 4.2.44).

4.2.46 Cartesian_coordinate_space_3d

A Cartesian_coordinate_space_3d is a coordinate space defined in three-dimensional space.

NOTE – Each three-dimensional geometric data is defined in a Cartesian_coordinate_space_3d.

A Cartesian_coordinate_space_3d is a type of Cartesian_coordinate_space (see 4.2.44).

4.2.47 Cartesian_point

A Cartesian_point is a point that is defined by its coordinates in a rectangular Cartesian coordinate system.

A Cartesian_point is a type of Point (see 4.2.322).

4.2.48 Catalog_thread

A Catalog_thread is a thread that is specified by a document containing the specification of the thread.

A Catalog_thread is a type of Thread_specification (see 4.2.458).

The data associated with a Catalog_thread are the following:

- is_specified_in.

The is_specified_in specifies the External_document (see 4.2.162) or the External_library_reference (see 4.2.165) that contains the thread specification.

See 4.3.179 and 4.3.180 for the application assertions.

There shall be exactly one object that the Catalog_thread is_specified_in.

4.2.49 Centre_of_mass

A Centre_of_mass is the centre of the mass of a body. The relative position of the Centre_of_mass within the body is an invariant datum relative to rotation and translation.

NOTE – Usually the Centre_of_mass is derivable from the shape representation. An example where this application object is additionally needed is the following: Sometimes the Centre_of_mass is calculated in some system at some point of time. But there are cases where the information (e.g. the Centre_of_mass of a die) is transferred into a following stage (e.g. feasibility check) where there is only a simplified shape representation that does not allow to calculate the exact Centre_of_mass. Yet in this stage the information about the Centre_of_mass is needed (e.g. to check whether the transport of a die with a crane works out).

A Centre_of_mass is a type of Shape_dependent_property (see 4.2.396).

The data associated with a Centre_of_mass are the following:

- centre_point;
- description;
- is_defined_in;
- value_interpretation.

4.2.49.1 centre_point

The centre_point specifies a point in three-dimensional space that defines the location of a Centre_of_mass in the Cartesian_coordinate_space (see 4.2.44).

4.2.49.2 description

The description specifies additional information about the Centre_of_mass.

See 4.3.182 for the application assertion.

The description need not be specified for a particular Centre_of_mass.

If present, there shall be exactly one object that defines the description for a Centre_of_mass.

4.2.49.3 is_defined_in

The is_defined_in specifies the Cartesian_coordinate_space (see 4.2.44) the Centre_of_mass is defined in.

See 4.3.181 for the application assertion.

4.2.49.4 value_interpretation

The value_interpretation specifies information on how the Centre_of_mass shall be interpreted. Where applicable the following values shall be used:

- 'required': The value_interpretation specifies the actual centre of mass is expected to be located at the given centre point;

- 'calculated': The value_interpretation specifies that the given centre point is calculated from a computerized description of the product;

- 'measured': The value_interpretation specifies that the given centre point is a statistical result of measurements made on actual products;

- 'estimated': The value_interpretation specifies that the given centre point is the result of an approximate judgement by a human being.

NOTE – The Centre_of_mass may be specified in the design stage of a product but it may also be documented as determined by measurements on prototypes.

4.2.50 Chamfer

A Chamfer is a transition between corresponding edges of two joining non-coplanar faces, having a linear cross section perpendicular to the intersection of the faces.

EXAMPLE 44 – Figure 10 depicts a Chamfer.

A Chamfer is a type of Transition_feature (see 4.2.476).

The data associated with a Chamfer are the following:

- is_defined_by.

The is_defined_by specifies the Chamfer_definition (see 4.2.52) that has the information defining the Chamfer.

See 4.3.183 for the application assertion.

4.2.51 Chamfer_angle

A Chamfer_angle is the definition of an angle used to specify a Chamfer (see 4.2.50).

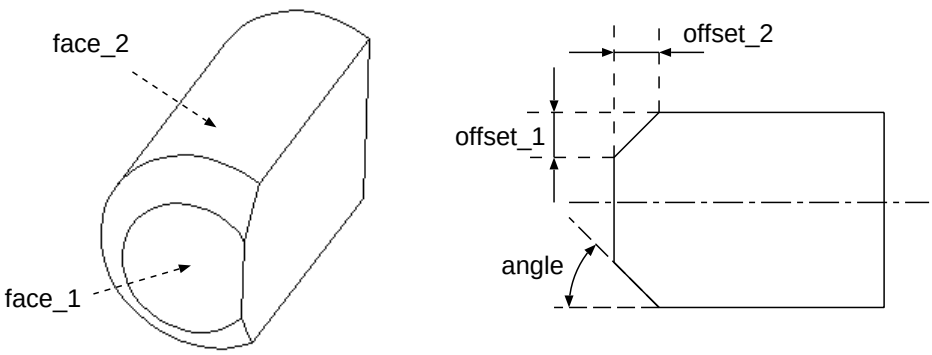


Figure 10 – Chamfer

The data associated with a Chamfer_angle are the following:

– angle.

The angle specifies the value of the angle. It is measured between a tangent of the concerned surface and a linear cross section where the plane on which both lines are lying is perpendicular to the intersection line of the faces and the point of tangency is lying on the intersection line.

See 4.3.184 for the application assertion.

4.2.52 Chamfer_definition

A Chamfer_definition is the set of parameters specifying a Chamfer (see 4.2.50).

The data associated with a Chamfer_definition are the following:

– offset_1;

– value_2.

4.2.52.1 offset_1

The offset_1 specifies the amount of offset measured from the intersection of the two faces along the first face.

NOTE – If the Chamfer_definition is used by a Round_hole (see 4.2.387) or by a Flat_with_chamfer_bottom (see 4.2.185) the second face is the face of the hole.

See 4.3.186 for the application assertion.

4.2.52.2 value_2

The value_2 specifies the second value that is necessary to define a Chamfer (see 4.2.50). In the case where the value_2 is a Chamfer_angle (see 4.2.51) the angle is measured from the second faces. In the case where the value_2 is a Chamfer_offset (see 4.2.53) the offset is measured from the intersection of the two faces along the second faces.

NOTE – If the Chamfer_definition is used by a Round_hole (see 4.2.387) or by a Flat_with_chamfer_bottom (see 4.2.185) the second face is the face of the hole.

See 4.3.185 and 4.3.186 for the application assertions.

There shall be exactly one object that defines the value_2 for a Chamfer_definition.

4.2.53 Chamfer_offset

A Chamfer_offset is the definition of an offset used to specify a Chamfer (see 4.2.50). The offset is measured as an arc length from the intersection curve of the two faces on the concerned face. The curve along which the arc length is measured is the intersection of a plane that is normal to the intersection curve of the two faces and the concerned face.

The data associated with a Chamfer_offset are the following:

- offset.

The offset specifies the value of the Chamfer_offset.

See 4.3.187 for the application assertion.

4.2.54 Change

A Change is a mechanism to collect the Model_change (see 4.2.283) objects and the Property_change (see 4.2.362) obects that describe the differences between the two objects referenced by the described relationship.

The data associated with a Change are the following:

- described_change;
- description.

4.2.54.1 described_change

The described_change specifies the relationship object that references a modified object and the corresponding original object.

See 4.3.188, 4.3.189, 4.3.191, 4.3.192, 4.3.193, 4.3.194, 4.3.195, 4.3.196, and 4.3.197 for the application assertions.

There shall be at least one object that defines the described_change for a Change.

4.2.54.2 description

The description specifies additional information about the Change.

See 4.3.190 for the application assertion.

The description need not be specified for a particular Change.

If present, there shall be exactly one object that defines the description for a Change.

4.2.55 Circle

A Circle is a planar unbounded curve where all points are equidistant from the centre of the Circle.

A Circle is a type of Curve (see 4.2.102).

4.2.56 Circular_boss

A Circular_boss is a protrusion feature that is specified by an enclosed volume bounded by a cylinder.

EXAMPLE 45 – Figure 11 illustrates a Circular_boss.

A Circular_boss is a type of Boss_feature (see 4.2.42).

The data associated with a Circular_boss are the following:

– boundary.

The boundary specifies the Circular_closed_profile (see 4.2.57) that defines the cross section of the Circular_boss.

See 4.3.198 for the application assertion.

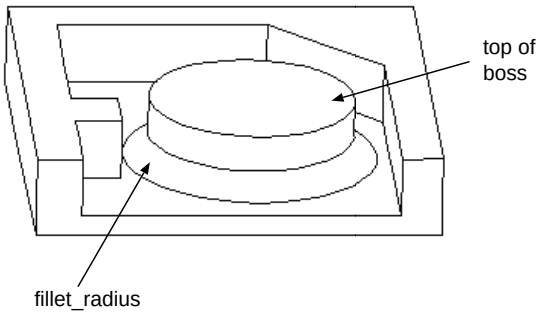


Figure 11 – Circular_boss

4.2.57 Circular_closed_profile

A Circular_closed_profile is specified by a circle that bounds an enclosed area.

EXAMPLE 46 – Figure 12 depicts a Circular_closed_profile.

A Circular_closed_profile is a type of Closed_profile (see 4.2.71).

The data associated with a Circular_closed_profile are the following:

– diameter.

The diameter specifies the distance across the Circular_closed_profile.

See 4.3.199 for the application assertion.

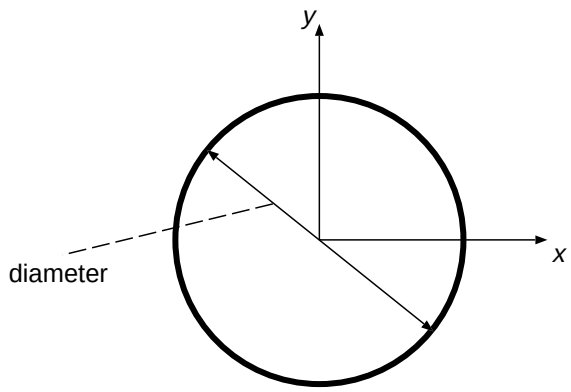


Figure 12 – Circular_closed_profile

4.2.58 Circular_offset_pattern

A Circular_offset_pattern is a modification of the placement of a particular occurrence of the base feature in a Circular_pattern (see 4.2.61) relative to its expected placement.

EXAMPLE 47 – Figure 13 depicts a Circular_offset_pattern.

The data associated with a Circular_offset_pattern are the following:

- angular_offset;
- index.

4.2.58.1 angular_offset

The angular_offset specifies the amount of offset from the expected feature location to the actual feature location along the diameter of the Circular_pattern (see 4.2.61) as an angular measurement.

4.2.58.2 index

The index specifies the number of the feature that has an offset.

4.2.59 Circular_omit_pattern

A Circular_omit_pattern is an omission of a particular occurrence of the base feature in a Circular_pattern (see 4.2.61).

EXAMPLE 48 – Figure 14 depicts a Circular_omit_pattern.

The data associated with a Circular_omit_pattern are the following:

- index.

The index specifies the number of the feature that is omitted.

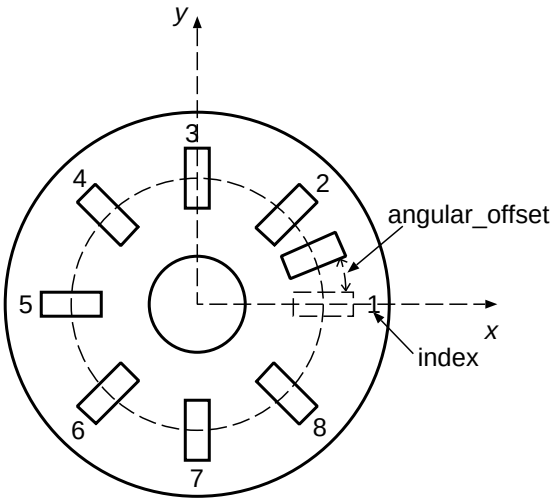


Figure 13 – Circular_offset_pattern

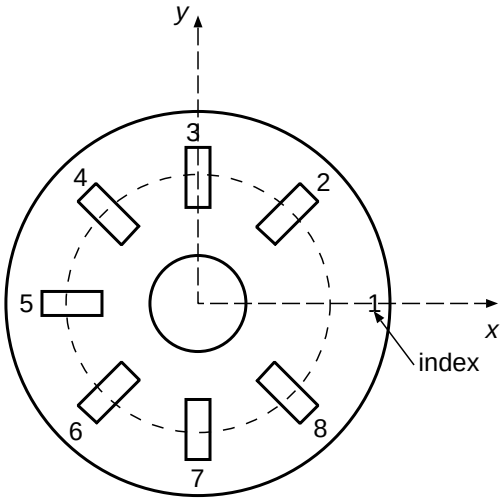


Figure 14 – Circular_omit_pattern

4.2.60 Circular_path

A Circular_path is specified by a direction of travel along an arc of constant radius around an axis.

A Circular_path is a type of Path (see 4.2.304).

Each Circular_path is either a Complete_circular_path (see 4.2.78) or a Partial_circular_path (see 4.2.302).

The data associated with a Circular_path are the following:

- radius;
- rotation_axis_defined_by.

4.2.60.1 radius

The radius specifies the radius of the Circular_path.

See 4.3.201 for the application assertion.

4.2.60.2 rotation_axis_defined_by

The rotation_axis_defined_by specifies an Axis_placement (see 4.2.33) that defines the location and orientation of the rotation axis which is the z-axis of the Axis_placement.

In the case where the Circular_path is used to define the sweep path of a profile, the x-z-plane of the Axis_placement shall coincide with the x-y-plane of the profile that is swept.

See 4.3.200 for the application assertion.

4.2.61 Circular_pattern

A Circular_pattern is specified by replications of a base feature where the replications are generated about a rotation axis. In the case where the Circular_pattern is applied as a Placed_feature (see 4.2.313) the rotation axis is the local z-axis. In the case the case where the Circular_pattern is applied as an Included_feature (see 4.2.232) the rotation axis is the z-axis of the Axis_placement (see 4.2.33) representing location and orientation of the pattern. The base feature's origin or the transformed base feature's origin is located at a distance from the local z-axis that equals half the diameter of the Circular_pattern.

NOTE – The base feature or the transformed base feature is the first instance of the Circular_pattern.

A Circular_pattern is a type of Replicate_feature (see 4.2.380).

The data associated with a Circular_pattern are the following:

- angular_spacing;
- diameter;

- missing_feature;
- number_of_features;
- relocated_feature;
- rotation.

4.2.61.1 angular_spacing

The angular_spacing specifies the angle between two instances in the Circular_pattern.

See 4.3.204 for the application assertion.

4.2.61.2 diameter

The diameter specifies twice the distance between the centre of the Circular_pattern and the location of the base feature.

NOTE – Usually the diameter can be determined from the location of the Circular_pattern and the location of the base feature. The attribute diameter may be instantiated e.g., to support process planning which may require the diameter value explicitly.

See 4.3.204 for the application assertion.

The diameter need not be specified for a particular Circular_pattern.

4.2.61.3 missing_feature

The missing_feature specifies those instances that are omitted in the Circular_pattern.

See 4.3.203 for the application assertion.

4.2.61.4 number_of_features

The number_of_features specifies the number of features used in the Circular_pattern.

4.2.61.5 relocated_feature

The relocated_feature specifies those instances that are relocated in the Circular_pattern.

See 4.3.202 for the application assertion.

4.2.61.6 rotation

The rotation specifies the angular measurement to rotate one feature with respect to the position of the previous feature in the Circular_pattern.

EXAMPLE 49 – Figure 15 depicts a Circular_pattern with a rotation specified that equals the negative angular_spacing of the Circular_pattern.

See 4.3.204 for the application assertion.

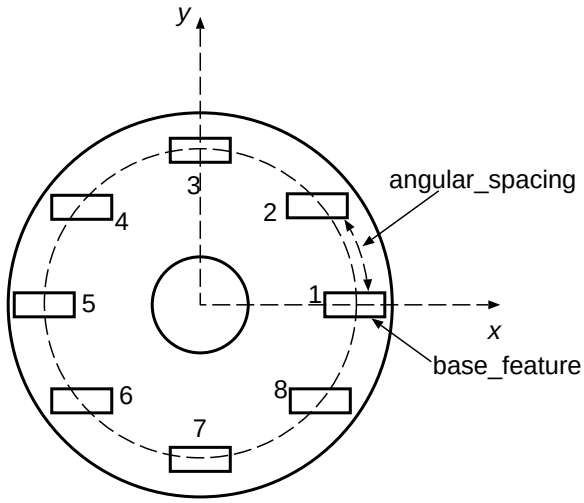


Figure 15 – Circular_pattern with rotation of base feature

4.2.62 Circular_runout_tolerance

A Circular_runout_tolerance is a mechanism to control circular elements of a part to a datum axis the part is rotated about. The tolerance is applied independently at any circular measuring position as the part is rotated 360 degrees. Where applied to surfaces constructed around a datum axis, Circular_runout_tolerance may be used to control the cumulative variations of circularity and coaxiality. Where applied to surfaces constructed at right angles to the datum axis, Circular_runout_tolerance controls circular elements of a plane surface.

A Circular_runout_tolerance is a type of Geometric_tolerance (see 4.2.216).

The data associated with a Circular_runout_tolerance are the following:

- angle;
- datum.

4.2.62.1 angle

The angle specifies the direction a runout tolerance is controlled in. If the angle is specified the runout tolerance applies in this angle which is fixed with respect to the datum axis. If the angle is not specified the runout tolerance applies normal to the feature surface.

The angle need not be specified for a particular Circular_runout_tolerance.

4.2.62.2 datum

The datum specifies the Datum_feature (see 4.2.117) or the Compound_datum (see 4.2.80) that define the reference for this type of geometric tolerance.

See 4.3.205 for the application assertion.

4.2.63 Circularity_tolerance

A Circularity_tolerance is the limitation of the amount of deviation points of a surface of revolution lying within the same plane perpendicular to a common axis may have from being equidistant from the axis. It defines a tolerance zone bounded by two concentric circles within which each element of the surface must lie. The Circularity_tolerance applies independently at any plane perpendicular to the common axis.

A Circularity_tolerance is a type of Geometric_tolerance (see 4.2.216).

4.2.64 Class_category_association

A Class_category_association is the assignment of a Specification_category (see 4.2.416) to a Product_class (see 4.2.348). Additionally this assignment specifies if the usage of one or more Specification (see 4.2.415) objects belonging to this Specification_category is mandatory or optional for all products of that Product_class.

NOTE 1 – In the case of a strict familiy concept for Specification_category objects this Class_category_association specifies that the usage of one Specification of this Specification_category is mandatory for all products of that Product_class. If no strict familiy concept for Specification_category objects is used this Class_category_association specifies that the usage of one or more Specification objects of this Specification_category is mandatory or optional for products of that Product_class.

NOTE 2 – The assignment of a Specification_category to a Product_class can not replace the association of single members of the Specification_category to the Product_class.

The data associated with a Class_category_association are the following:

- category;
- mandatory;
- product_class.

4.2.64.1 category

The category specifies the Specification_category (see 4.2.416) that is associated with the Product_class (see 4.2.348).

See 4.3.207 for the application assertion.

4.2.64.2 mandatory

The mandatory specifies whether one or more Specification (see 4.2.415) objects of the associated Specification_category (see 4.2.416) have to be used or may be used (optional) for products within the referenced Product_class (see 4.2.348).

EXAMPLE 50 – The specification category 'radio' may be associated 'optional' to the product class of a car; the specification category 'engine' is an example for a 'mandatory' association.

4.2.64.3 product_class

The product_class specifies the Product_class (see 4.2.348) for which the Specification_category (see 4.2.416) is valid.

See 4.3.206 for the application assertion.

4.2.65 Class_condition_association

A Class_condition_association is a relationship between a Specification_expression (see 4.2.417) and a Product_class (see 4.2.348). This relationship contains the information that a particular Specification_expression is valid for all products of that Product_class.

The data associated with a Class_condition_association are the following:

- condition;
- condition_type;
- description;
- product_class.

4.2.65.1 condition

The condition specifies the Specification_expression (see 4.2.417) that is assigned to the Product_class (see 4.2.348).

See 4.3.210 for the application assertion.

4.2.65.2 condition_type

The condition_type specifies the meaning of the association. The following values shall be used where applicable:

- 'part usage': The Specification_expression (see 4.2.417) specifies a condition for the usage of the components of an Item_solution (see 4.2.246) in the products of the associated Product_class (see 4.2.348). In this case the Class_condition_association shall be referenced by at least one Configuration (see 4.2.89) object;

– 'identification': The Specification_expression specifies a condition that enables to distinguish the associated Product_class from other Product_class objects. For a top level node in a hierarchy of Product_class objects this value is not applicable. This identification is part of the identification of all sub classes of this Product_class;

– 'validity': The Specification_expression (see 4.2.417) specifies a condition that is used to verify a given set of Specification (see 4.2.415) objects for the associated Product_class. That means that the Specification_expression evaluates to true if the set of Specification objects is valid; otherwise it evaluates to false.

4.2.65.3 description

The description specifies additional information about the Class_condition_association.

See 4.3.208 for the application assertion.

The description need not be specified for a particular Class_condition_association.

If present, there shall be exactly one object that defines the description for a Class_condition_association.

4.2.65.4 product_class

The product_class specifies the Product_class (see 4.2.348) for which the Specification_expression (see 4.2.417) is valid.

See 4.3.209 for the application assertion.

4.2.66 Class_function_association

A Class_function_association is a relationship between a Product_function (see 4.2.352) and a Product_class (see 4.2.348).

NOTE – The semantic of this relationship is specified by the attribute relation_type.

The data associated with a Class_function_association are the following:

– description;

– product_class;

– product_function;

– relation_type.

4.2.66.1 description

The description specifies additional information about the Class_function_association.

See 4.3.211 for the application assertion.

The description need not be specified for a particular Class_function_association.

If present, there shall be exactly one object that defines the description for a Class_function_-association.

4.2.66.2 product_class

The product_class specifies the Product_class (see 4.2.348) that is associated with the Product_-function (see 4.2.352).

See 4.3.212 for the application assertion.

4.2.66.3 product_function

The product_function specifies the Product_function (see 4.2.352) that is associated with the Product_class (see 4.2.348).

See 4.3.213 for the application assertion.

4.2.66.4 relation_type

The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:

- 'possibility': Performing the Product_function (see 4.2.352) has to be considered for at least some products of the Product_class (see 4.2.348);

- 'main function': The Product_function is one of the primary functions performed by each product that is a member of the Product_class.

4.2.67 Class_inclusion_association

A Class_inclusion_association is the assignment of a Specification_inclusion (see 4.2.418) to a Product_class (see 4.2.348). This assignment contains the information that a particular Specification_inclusion applies for all products of that Product_class.

The data associated with a Class_inclusion_association are the following:

- description;

- inclusion;

- product_class.

4.2.67.1 description

The description specifies additional information about the Class_inclusion_association.

See 4.3.214 for the application assertion.

The description need not be specified for a particular Class_inclusion_association.

If present, there shall be exactly one object that defines the description for a Class_inclusion_association.

4.2.67.2 inclusion

The inclusion specifies the Specification_inclusion (see 4.2.418) that is associated with the Product_class (see 4.2.348).

See 4.3.216 for the application assertion.

4.2.67.3 product_class

The product_class specifies the Product_class (see 4.2.348) for which the Specification_inclusion (see 4.2.418) is valid.

See 4.3.215 for the application assertion.

4.2.68 Class_specification_association

A Class_specification_association is a relationship between a Specification (see 4.2.415) and a Product_class (see 4.2.348). This Specification serves as a potential characteristic of all products belonging to the Product_class.

The data associated with a Class_specification_association are the following:

- association_type;
- product_class;
- specification.

4.2.68.1 association_type

The association_type specifies the kind of availability of a particular Specification in a Product_class (see 4.2.348). Where applicable the following values shall be used:

- 'replaceable standard': The Specification is a default characteristic of the products belonging to the Product_class as long as no other specification of the same Specification_category is chosen;

  EXAMPLE 51 – 'power brake' is a 'replaceable standard' for Golf; it can be replaced by 'ABS' which is another specification of the same specification category 'brake system'.

- 'non replaceable standard': The Specification is a characteristic of any product belonging to the Product_class;

  EXAMPLE 52 – 'South East Asia climate zone' is a 'non replaceable standard' for a Sunny with market context 'Japan'.

– 'availability': The Specification is a potential characteristic of any product belonging to a high level Product_class. It is not specified if this is an option or a standard;

EXAMPLE 53 – 'front wheel drive' or 'four wheel drive' are available for a product class 'Voyager'.

– 'identification': The Specification is a characteristic that enables to distinguish the associated Product_class from other Product_class objects. This is a kind of 'non replaceable standard'. For a top level node in a hierarchy of Product_class objects this value shall not be applied. This identification is part of the identification of all sub classes of this Product_class;

EXAMPLE 54 – The car body type 'limousine' is identifying for product class 'S40'.

– 'option': The Specification is a characteristic of a product if explicitly chosen. The Specification replaces another Specification of the same specification category if the replaced Specification is associated with the Product_class as 'replaceable standard';

EXAMPLE 55 – 'sun roof' or 'radio with CD' are common options proposed to the customer.

4.2.68.2 product_class

The product_class specifies the Product_class (see 4.2.348) for which the Specification (see 4.2.415) is valid.

See 4.3.217 for the application assertion.

4.2.68.3 specification

The specification specifies the Specification (see 4.2.415) that is associated with the Product_class (see 4.2.348).

See 4.3.218 for the application assertion.

4.2.69 Classification_attribute

A Classification_attribute is a characteristic used to classify an Item (see 4.2.236) associated with the corresponding General_classification (see 4.2.192).

The data associated with a Classification_attribute are the following:

– attribute_source;

– attribute_value;

– description;

– id;

– name.

4.2.69.1 attribute_source

The attribute_source specifies the External_document (see 4.2.162) or the External_library_-reference (see 4.2.165) that contains the specification of the Classification_attribute. The External_library_reference (see 4.2.165) specifies the unique identification of a property, i.e. a property_BSU according to ISO 13584–42, for an ISO 13584 compliant library.

See 4.3.219 and 4.3.220 for the application assertions.

The attribute_source need not be specified for a particular Classification_attribute.

If present, there shall be exactly one object that defines the attribute_source for a Classification_-attribute.

4.2.69.2 attribute_value

The attribute_value specifies the Property_value (see 4.2.364) that represents the value of the Classification_attribute. The combination of 'name' and 'value_interpretation' of the referenced Property_value shall be unique within the scope of a Classification_attribute.

See 4.3.222 for the application assertion.

4.2.69.3 description

The description specifies additional information about the Classification_attribute.

See 4.3.221 for the application assertion.

The description need not be specified for a particular Classification_attribute.

If present, there shall be exactly one object that defines the description for a Classification_-attribute.

4.2.69.4 id

The id specifies the identifier of the Classification_attribute that shall be unique within the scope of the associated General_classification (see 4.2.192).

EXAMPLE 56 – 'a' or 'b' for length or width attributes or 'r' for radius attributes are examples for identifiers of Classification_attribute objects. The meaning of such ids is usually specified in external sources.

4.2.69.5 name

The name specifies the word or group of words by which the Classification_attribute is referred to.

EXAMPLE 57 – 'length', 'width', or 'radius' are examples for the name of a Classification_attribute.

See 4.3.221 for the application assertion.

The name need not be specified for a particular Classification_attribute.

If present, there shall be exactly one object that defines the name for a Classification_attribute.

4.2.70 Classification_system

A Classification_system is a method for classifying objects

NOTE – Each object that is assigned to a Classification_system is one element of the Classification_system. The complete set of possible values is usually described in an external source.

The data associated with a Classification_system are the following:

- classification_system_source;
- description;
- id;
- name.

4.2.70.1 classification_system_source

The classification_system_source specifies the External_document (see 4.2.162) or the External_library_reference (see 4.2.165) that contains the specification for the Classification_system. The External_library_reference specifies the unique identification of a supplier or class, i.e. a supplier_BSU or class_BSU according to ISO 13584–42, for an ISO 13584 compliant library.

See 4.3.223 and 4.3.224 for the application assertions.

The classification_system_source need not be specified for a particular Classification_system.

If present, there shall be exactly one object that defines the classification_system_source for a Classification_system.

4.2.70.2 description

The description specifies additional information about the Classification_system.

See 4.3.225 for the application assertion.

The description need not be specified for a particular Classification_system.

If present, there shall be exactly one object that defines the description for a Classification_system.

4.2.70.3 id

The id specifies the identifier of the Classification_system.

The id need not be specified for a particular Classification_system.

4.2.70.4 name

The name specifies the word or group of words by which the Classification_system is referred to.

See 4.3.225 for the application assertion.

There shall be exactly one object that defines the name for a Classification_system.

4.2.71 Closed_profile

A Closed_profile is specified by an outline that bounds an enclosed area with no opening.

A Closed_profile is a type of Profile (see 4.2.355).

Each Closed_profile is either a Ngon_profile (see 4.2.288), a Rectangular_closed_profile (see 4.2.370), a General_closed_profile (see 4.2.194), or a Circular_closed_profile (see 4.2.57).

4.2.72 Closed_profile_usage

A Closed_profile_usage is the identification of a Closed_profile (see 4.2.71) and possibly of a geometric Transformation (see 4.2.472) used to place the Closed_profile in a reference coordinate space.

The data associated with a Closed_profile_usage are the following:

- placement;
- profile.

4.2.72.1 placement

The placement specifies the geometric Transformation (see 4.2.472) that locates and orients the Closed_profile in the reference coordinate space.

See 4.3.227 for the application assertion.

The placement need not be specified for a particular Closed_profile_usage.

4.2.72.2 profile

The profile specifies the Closed_profile (see 4.2.71) that is used.

See 4.3.226 for the application assertion.

4.2.73 Closed_shell

A Closed_shell is a collection of at least one Face (see 4.2.167) that bounds a region in the three-dimensional space. The topological normal of the shell is defined as being directed from the finite to the infinite region. A void is represented by an interior shell that builds a "hole"

inside an outer shell. For such an interior shell the topological normal will point into the solid material. A Closed_shell shall not self-intersect.

The data associated with a Closed_shell are the following:

- shell_boundary.

The shell_boundary specifies the Face (see 4.2.167) that represents the boundary of a Closed_shell (see 4.2.73).

See 4.3.228 for the application assertion.

4.2.74 Coating_layer

A Coating_layer is one layer of covering applied to a surface

A Coating_layer is a type of Surface_condition (see 4.2.437).

The data associated with a Coating_layer are the following:

- thickness;
- used_material.

4.2.74.1 thickness

The thickness specifies the thickness of the Coating_layer.

See 4.3.230 for the application assertion.

The thickness need not be specified for a particular Coating_layer.

4.2.74.2 used_material

The used_material specifies the material that is used for the Coating_layer.

See 4.3.229 for the application assertion.

4.2.75 Coating_layer_relationship

A Coating_layer_relationship is the relationship between two Coating_layer (see 4.2.74) objects that are applied to the same surface. A sequence of the coating layers can be built up by using multiple Coating_layer_relationship objects.

The data associated with a Coating_layer_relationship are the following:

- description;
- following_layer;
- id;
- preceding_layer;

– purpose.

4.2.75.1 description

The description specifies any technical conditions or dependencies that have to be considered if the coating layers are applied in the defined sequence.

EXAMPLE 58 – Two coating layers of paint might be specified in a sequence with the condition that the first layer has to be dried out for at least 20 minutes before the second layer of paint can be applied.

See 4.3.232 for the application assertion.

The description need not be specified for a particular Coating_layer_relationship.

If present, there shall be exactly one object that defines the description for a Coating_layer_-relationship.

4.2.75.2 following_layer

The following_layer specifies the coating layer that is applied subsequently, i.e., the coating layer that is applied to the surface after the preceding_layer has been applied.

See 4.3.231 for the application assertion.

4.2.75.3 id

The id specifies the identifier of the Coating_layer_relationship.

4.2.75.4 preceding_layer

The preceding_layer specifies the coating layer that is applied previously, i.e., the coating layer that has been applied to the surface before the following_layer is applied.

See 4.3.231 for the application assertion.

4.2.75.5 purpose

The purpose specifies the rationale of the Coating_layer_relationship.

See 4.3.232 for the application assertion.

The purpose need not be specified for a particular Coating_layer_relationship.

If present, there shall be exactly one object that defines the purpose for a Coating_layer_-relationship.

4.2.76 Collected_item_association

A Collected_item_association is a mechanism to associate Item_instance (see 4.2.241) objects with a Collection_definition (see 4.2.77).

A Collected_item_association is a type of Item_definition_instance_relationship (see 4.2.239).

The data associated with a Collected_item_association are the following:

- relating.

The relating specifies the Collection_definition (see 4.2.77) that collects the Design_discipline_item_definition (see 4.2.134) objects.

See 4.3.233 for the application assertion.

4.2.77 Collection_definition

A Collection_definition is the definition of an Item_version (see 4.2.248) that serves as a collector for Item_instance (see 4.2.241) objects that are mounted in the same vehicle but may not be assembled together.

EXAMPLE 59 – Brake pads on left and right front brake from same supplier, locks and keys of the car all from one supplier, wooden panels in interior from same supplier (to ensure common structure of the wood) are examples for collections.

A Collection_definition is a type of Design_discipline_item_definition (see 4.2.134).

The data associated with a Collection_definition are the following:

- purpose.

The purpose specifies the rationale behind the Collection_definition.

See 4.3.234 for the application assertion.

The purpose need not be specified for a particular Collection_definition.

If present, there shall be exactly one object that defines the purpose for a Collection_definition.

4.2.78 Complete_circular_path

A Complete_circular_path is specified by a direction of travel that begins and ends at the same point on the arc.

EXAMPLE 60 – Figure 16 illustrates a Complete_circular_path.

A Complete_circular_path is a type of Circular_path (see 4.2.60).

4.2.79 Composite_curve

A Composite_curve is a curve that is composed of a number of curves which join end to end. A Composite_curve may be closed and self-intersecting.

A Composite_curve is a type of Curve (see 4.2.102).

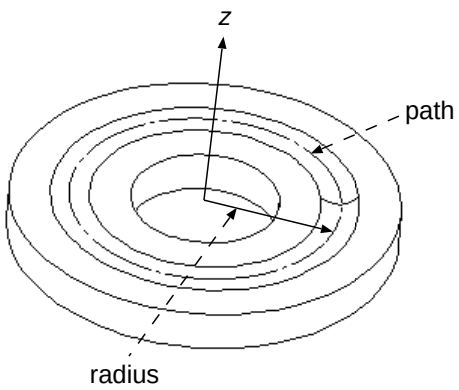


Figure 16 – Complete_circular_path

4.2.80 Compound_datum

A Compound_datum is a set of two or more Datum_feature (see 4.2.117) objects of equal importance which are used to establish a single datum plane or axis.

A Compound_datum is a type of Datum (see 4.2.116).

The data associated with a Compound_datum are the following:

– is_made_up_by.

The is_made_up_by specifies the set of Datum_feature (see 4.2.117) objects that form the Compound_datum.

See 4.3.235 for the application assertion.

4.2.81 Compound_feature_in_panel

A Compound_feature_in_panel is a union of Process_feature_in_panel (see 4.2.334) objects that allows the definition of complex features based on existing Process_feature_in_panel definitions.

In the case where the Compound_feature_in_panel is applied as a Placed_feature (see 4.2.313) all of its elements except for Transition_feature (see 4.2.476) objects shall be defined in their own local coordinate spaces and then transformed into the coordinate space of the Compound_feature_in_panel. In the case where the Compound_feature_in_panel is applied as an Included_feature (see 4.2.232) the Compound_feature_in_panel as well as all of its elements shall be defined in the part's coordinate space.

A Compound_feature_in_panel is a type of Process_feature_in_panel (see 4.2.334).

The data associated with a Compound_feature_in_panel are the following:

– feature_type.

The feature_type specifies the kind of Compound_feature_in_panel.

4.2.82 Compound_feature_in_panel_element

A Compound_feature_in_panel_element is a feature element that is used in the definition of a Compound_feature_in_panel (see 4.2.81).

The data associated with a Compound_feature_in_panel_element are the following:

- element_placement;
- feature;
- is_element_for.

4.2.82.1 element_placement

The element_placement specifies the geometric Transformation (see 4.2.472) that is used to place the Compound_feature_in_panel_element in the coordinate space of the Compound_feature_in_panel (see 4.2.81) in the case where the Compound_feature_in_panel is applied as a Placed_feature (see 4.2.313). In the case where the Compound_feature_in_panel is applied as an Included_feature (see 4.2.232) the element_placement shall not be instantiated.

See 4.3.238 for the application assertion.

The element_placement need not be specified for a particular Compound_feature_in_panel_element.

4.2.82.2 feature

The feature specifies the Process_feature_in_panel (see 4.2.334) that is the feature definition of the Compound_feature_in_panel_element.

See 4.3.237 and 4.3.239 for the application assertions.

There shall be exactly one object that defines the feature for a Compound_feature_in_panel_element.

4.2.82.3 is_element_for

The is_element_for specifies the Compound_feature_in_panel (see 4.2.81) the Compound_feature_in_panel_element is a component of.

See 4.3.236 for the application assertion.

4.2.83 Compound_feature_in_panel_element_relationship

A Compound_feature_in_panel_element_relationship is a mechanism to define the sequence in which the Compound_feature_in_panel_element (see 4.2.82) elements are applied in the Compound_feature_in_panel (see 4.2.81).

NOTE – The sequence defined by the Compound_feature_in_panel_element_relationship need not be the same as the manufacturing sequence of the Compound_feature_in_panel.

The data associated with a Compound_feature_in_panel_element_relationship are the following:

- predecessor;
- successor.

4.2.83.1 predecessor

The predecessor specifies the Compound_feature_in_panel_element (see 4.2.82) that is preceding the other feature in the Compound_feature_in_panel_element_relationship.

See 4.3.240 for the application assertion.

4.2.83.2 successor

The successor specifies the Compound_feature_in_panel_element (see 4.2.82) that is succeeding the other feature in the Compound_feature_in_panel_element_relationship.

See 4.3.240 for the application assertion.

4.2.84 Compound_feature_in_solid

A Compound_feature_in_solid is a union of Process_feature_in_solid (see 4.2.335) objects that allows the definition of complex features based on existing Process_feature_in_solid definitions.

In the case where the Compound_feature_in_solid is applied as a Placed_feature (see 4.2.313) all of its elements except for Transition_feature (see 4.2.476) or Thread_feature (see 4.2.457) objects shall be defined in their own local coordinate spaces and then transformed into the coordinate space of the Compound_feature_in_solid. In the case where the Compound_feature_in_solid is applied as an Included_feature (see 4.2.232) the Compound_feature_in_solid as well as all of its elements shall be defined in the part's coordinate space.

A Compound_feature_in_solid is a type of Process_feature_in_solid (see 4.2.335).

The data associated with a Compound_feature_in_solid are the following:

- feature_type.

The feature_type specifies the kind of Compound_feature_in_solid.

EXAMPLE 61 – 'counterbore hole' or 'countersunk hole' are examples for feature_type.

4.2.85 Compound_feature_in_solid_element

A Compound_feature_in_solid_element is the feature element that is used in the definition of a Compound_feature_in_solid (see 4.2.84). When the Compound_feature_in_solid_element is a Thread_feature (see 4.2.457) there shall be a Compound_feature_in_solid_element_relationship (see 4.2.86) that references the Compound_feature_in_solid_element representing the Thread_feature as successor and a Compound_feature_in_solid_element representing either a Circular_boss (see 4.2.56) or a Round_hole (see 4.2.387) as predecessor.

The data associated with a Compound_feature_in_solid_element are the following:

- element_placement;
- feature;
- is_element_for.

4.2.85.1 element_placement

The element_placement specifies the geometric Transformation (see 4.2.472) that is used to place the Compound_feature_in_solid_element in the coordinate space of the Compound_feature_in_solid (see 4.2.84) in the case where the Compound_feature_in_solid is applied as a Placed_feature (see 4.2.313); the element_placement shall not be used if the Compound_feature_in_solid_element is a Thread_feature (see 4.2.457) or a Transition_feature (see 4.2.476). In the case where the Compound_feature_in_solid is applied as an Included_feature (see 4.2.232) the element_placement shall not be instantiated.

See 4.3.244 for the application assertion.

The element_placement need not be specified for a particular Compound_feature_in_solid_element.

4.2.85.2 feature

The feature specifies the Process_feature_in_solid (see 4.2.335) that is the feature definition of the Compound_feature_in_solid_element.

See 4.3.242, 4.3.243, and 4.3.245 for the application assertions.

There shall be exactly one object that defines the feature for a Compound_feature_in_solid_element.

4.2.85.3 is_element_for

The is_element_for specifies the Compound_feature_in_solid_element (see 4.2.85) the Compound_feature_in_solid_element is a component of.

See 4.3.241 for the application assertion.

4.2.86 Compound_feature_in_solid_element_relationship

A Compound_feature_in_solid_element_relationship is a mechanism to define the sequence in which the Compound_feature_in_solid_element (see 4.2.85) elements are applied in the Compound_feature_in_solid (see 4.2.84).

NOTE – The sequence defined by the Compound_feature_in_solid_element_relationship need not be the same as the manufacturing sequence of the Compound_feature_in_solid.

The data associated with a Compound_feature_in_solid_element_relationship are the following:

- predecessor;
- successor.

4.2.86.1 predecessor

The predecessor specifies the Compound_feature_in_solid_element (see 4.2.85) that is preceding the other feature in the Compound_feature_in_solid_element_relationship.

See 4.3.246 for the application assertion.

4.2.86.2 successor

The successor specifies the Compound_feature_in_solid_element (see 4.2.85) that is succeeding the other feature in the Compound_feature_in_solid_element_relationship.

See 4.3.246 for the application assertion.

4.2.87 Compound_model

A Compound_model is a model where elements of different dimensionality such as wireframes or surface model are topologically connected. The topology of a Compound_model may be non-manifold.

NOTE – A Compound_model may include non-manifold surface models as well as topologically bounded three-dimensional wireframe models.

A Compound_model is a type of Geometric_model (see 4.2.211).

The data associated with a Compound_model are the following:

- element.

The element specifies the topologically connected elements that form the Compound_model.

See 4.3.247 for the application assertion.

4.2.88 Concentricity_tolerance

A Concentricity_tolerance is a mechanism to control the condition where the axes of all cross-sectional elements of a surface of revolution are common to the axis of a datum feature. It specifies a cylindrical tolerance zone whose axis coincides with a datum axis and within which all cross-sectional axes of the feature being controlled must lie.

A Concentricity_tolerance is a type of Geometric_tolerance (see 4.2.216).

The data associated with a Concentricity_tolerance are the following:

- datum.

The datum specifies the Datum_feature (see 4.2.117) or the Compound_datum (see 4.2.80) that defines the reference for this type of geometric tolerance.

See 4.3.248 for the application assertion.

4.2.89 Configuration

A Configuration is the association of a Specification_expression (see 4.2.417) or a Specification (see 4.2.415) with an Item_solution (see 4.2.246) or with a Process_operation (see 4.2.336) in the context of a certain Product_class (see 4.2.348). The association may be limited by time, by serial number or by lot size.

Each Configuration may be one of the following: a Serial_configuration (see 4.2.393), a Lot_-configuration (see 4.2.272), or a Dated_configuration (see 4.2.115).

The data associated with a Configuration are the following:

- is_solution_for;
- usage.

4.2.89.1 is_solution_for

The is_solution_for specifies the characteristic or combination of characteristics for which the Item_solution (see 4.2.246) provides a technical solution or which is needed to control a Process_-operation (see 4.2.336). These characteristics are defined by a Class_specification_association (see 4.2.68) and combinations of characteristics are defined by a Class_condition_association (see 4.2.65) where the attribute 'condition type' is 'part usage'.

See 4.3.249 and 4.3.250 for the application assertions.

There shall be exactly one object that the Configuration is_solution_for.

4.2.89.2 usage

The usage specifies an Item_solution (see 4.2.246) or a Process_operation (see 4.2.336) that is controlled by a Configuration.

See 4.3.251 and 4.3.252 for the application assertions.

There shall be exactly one object that defines the usage for a Configuration.

4.2.90 Conical_bottom

A Conical_bottom is a hole bottom that is shaped like a cone. A Conical_bottom is characterized by a constant decrease in the hole diameter until the diameter is zero or until the surface becomes tangent to a specified tip_radius.

A Conical_bottom is a type of Hole_bottom_condition (see 4.2.228).

The data associated with a Conical_bottom are the following:

- tip_angle;
- tip_radius.

4.2.90.1 tip_angle

The tip_angle specifies the opening angle of the cone. The tip_angle shall have a value greater than 0 degrees and smaller than 180 degrees.

See 4.3.253 for the application assertion.

4.2.90.2 tip_radius

The tip_radius specifies the radius at the tip of the cone.

See 4.3.253 for the application assertion.

The tip_radius need not be specified for a particular Conical_bottom.

4.2.91 Conical_surface

A Conical_surface is a surface that could be produced by revolving a line in three-dimensional geometric space about any intersecting line. The angle between the axis and the revolved line must be between and not equal to 0 and 90 degrees. A Conical_surface shall be trimmed to a finite extension if used in a Surface_model (see 4.2.439) or a Geometrically_bounded_surface_-model (see 4.2.220).

A Conical_surface is a type of Surface (see 4.2.434).

4.2.92 Constant_radius_edge_round

A Constant_radius_edge_round is an Edge_round (see 4.2.154) with a constant radius.

A Constant_radius_edge_round is a type of Edge_round (see 4.2.154).

The data associated with a Constant_radius_edge_round are the following:

- first_offset;
- radius;
- second_offset.

4.2.92.1 first_offset

The first_offset specifies the amount of offset from the first Face (see 4.2.167) that defines the location of tangency.

EXAMPLE 62 – Figure 17 illustrates the use of offsets with a Constant_radius_edge_round.

NOTE – The use of offsets causes the location of tangency to be 'outside the material' and not on the part's surface.

See 4.3.254 for the application assertion.

The first_offset need not be specified for a particular Constant_radius_edge_round.

4.2.92.2 radius

The radius specifies the amount of curvature for a convex transition representing a Constant_-radius_edge_round.

See 4.3.254 for the application assertion.

4.2.92.3 second_offset

The second_offset specifies the amount of offset from the second Face (see 4.2.167) that defines the location of tangency.

See 4.3.254 for the application assertion.

The second_offset need not be specified for a particular Constant_radius_edge_round.

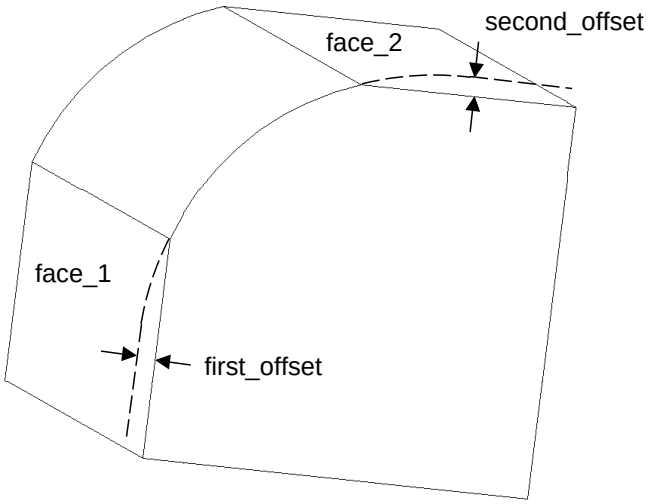


Figure 17 – Constant_radius_edge_round with offsets

4.2.93 Constant_radius_fillet

A Constant_radius_fillet is a Fillet (see 4.2.181) with a constant radius.

A Constant_radius_fillet is a type of Fillet (see 4.2.181).

The data associated with a Constant_radius_fillet are the following:

- first_offset;
- radius;
- second_offset.

4.2.93.1 first_offset

The first_offset specifies the amount of offset from the first Face (see 4.2.167) that defines the location of tangency.

EXAMPLE 63 – Figure 18 illustrates the use of offsets with a Constant_radius_fillet.

NOTE – The use of offsets causes the location of tangency to be 'inside the material' and not on the part's surface.

See 4.3.255 for the application assertion.

The first_offset need not be specified for a particular Constant_radius_fillet.

4.2.93.2 radius

The radius specifies the amount of curvature for a concave transition representing a Constant_-radius_fillet.

See 4.3.255 for the application assertion.

4.2.93.3 second_offset

The second_offset specifies the amount of offset from the second Face (see 4.2.167) that defines the location of tangency.

See 4.3.255 for the application assertion.

The second_offset need not be specified for a particular Constant_radius_fillet.

4.2.94 Constructive_geometric_element

A Constructive_geometric_element is an element of a Constructive_geometry (see 4.2.95).

The data associated with a Constructive_geometric_element are the following:

- description;
- element.

4.2.94.1 description

The description specifies the meaning of the Constructive_geometric_element.

EXAMPLE 64 – The description may be e.g., 'rotation axis' or 'mirroring plane'.

The description need not be specified for a particular Constructive_geometric_element.

4.2.94.2 element

The element specifies the geometric element that serves as a Constructive_geometric_element.

See 4.3.256, 4.3.257, 4.3.258, 4.3.259, 4.3.260, 4.3.261, and 4.3.262 for the application assertions.

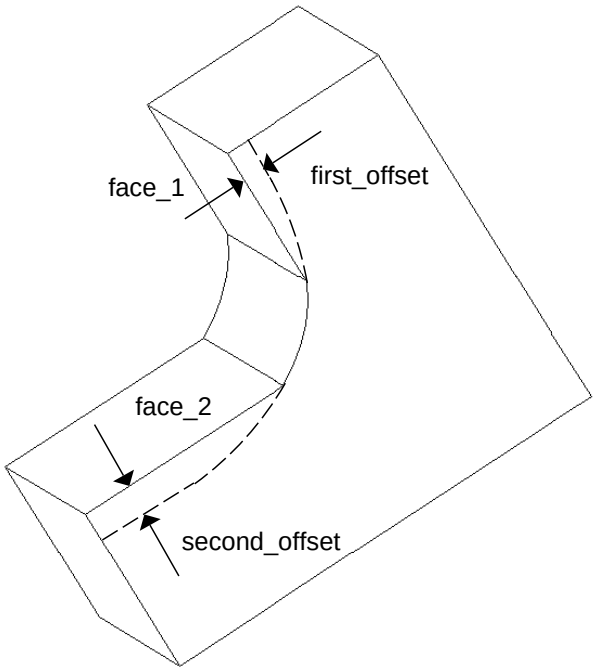


Figure 18 – Constant_radius_fillet with offsets

There shall be exactly one object that defines the element for a Constructive_geometric_element.

4.2.95 Constructive_geometry

A Constructive_geometry is a collection of non-shape geometry that is used to support the definition of the geometry that is representing the shape of the part.

The data associated with a Constructive_geometry are the following:

- element.

The element specifies the Constructive_geometric_element (see 4.2.94) objects that the Constructive_geometry consists of.

See 4.3.263 for the application assertion.

4.2.96 Constructive_geometry_association

A Constructive_geometry_association is a mechanism to associate the geometry representing the part's shape with the constructive geometry which was used for its definition.

The data associated with a Constructive_geometry_association are the following:

- base_geometry;

- constructive_geometry;

- description.

4.2.96.1 base_geometry

The base_geometry specifies the Geometric_model (see 4.2.211) the Constructive_geometry (see 4.2.95) is used in.

See 4.3.264 and 4.3.265 for the application assertions.

There shall be exactly one object that defines the base_geometry for a Constructive_geometry_-association.

4.2.96.2 constructive_geometry

The constructive_geometry specifies the Constructive_geometry (see 4.2.95) that supports the definition of the geometry representing the shape of the part.

See 4.3.264 for the application assertion.

4.2.96.3 description

The description specifies additional information about the Constructive_geometry_association.

See 4.3.266 for the application assertion.

The description need not be specified for a particular Constructive_geometry_association.

If present, there shall be exactly one object that defines the description for a Constructive_-geometry_association.

4.2.97 Contact_ratio

A Contact_ratio is the measure of the quality of contact of a shape on a reference surface. It is expressed as the ratio between the measure of the area of contact and the measure of the area of the reference surface.

A Contact_ratio is a type of Surface_condition (see 4.2.437).

The data associated with a Contact_ratio are the following:

- measuring_reference;
- ratio_value.

4.2.97.1 measuring_reference

The measuring_reference specifies a Physical_model (see 4.2.312) that serves as the reference for the determination of a Contact_ratio.

See 4.3.267 for the application assertion.

4.2.97.2 ratio_value

The ratio_value specifies a value between 0.0 and 1.0 that is the ratio between the measure of the area of contact and the measure of the area of the reference surface.

4.2.98 Context_dependent_appearance_assignment

A Context_dependent_appearance_assignment is the appearance of geometric elements and annotation with regard to a context in which these elements are represented. The context is defined by a Drawing_sheet (see 4.2.152), Group (see 4.2.222), Layer (see 4.2.261), Model_image (see 4.2.284), View (see 4.2.490), or View_area (see 4.2.491). The Context_dependent_appearance_assignment is overridden by appearances associated to elements by an Appearance_assignment (see 4.2.19).

A Context_dependent_appearance_assignment is a type of Appearance_assignment (see 4.2.19).

The data associated with a Context_dependent_appearance_assignment are the following:

- appearance_context.

The appearance_context specifies the context where the appearance components are valid.

See 4.3.268, 4.3.269, 4.3.270, 4.3.271, 4.3.272, and 4.3.273 for the application assertions.

There shall be exactly one object that defines the appearance_context for a Context_dependent_appearance_assignment.

4.2.99 Contract

A Contract is a binding agreement concerning the design of Item_version (see 4.2.248) objects and/or the carrying out of Activity (see 4.2.2) objects.

The data associated with a Contract are the following:

- contracted_element;
- id;
- ordered_price.

4.2.99.1 contracted_element

The contracted_element specifies the object that is subject of the Contract.

See 4.3.274 and 4.3.275 for the application assertions.

4.2.99.2 id

The id specifies the identifier of the Contract that shall be unique within the scope of an organization.

4.2.99.3 ordered_price

The ordered_price specifies the price for an Item_version (see 4.2.248) object or an Activity (see 4.2.2) object according to the underlying Contract.

The ordered_price need not be specified for a particular Contract.

4.2.100 Csg_model

A Csg_model is a collection of solids combined using regularised Boolean operations according to a tree (directed acyclic graph) with CSG primitives and non primitive solid models at the leaves and Boolean operations (and their associated Boolean results) at the nodes. A Csg_model is the root of the tree.

A Csg_model is a type of Geometric_model (see 4.2.211).

The data associated with a Csg_model are the following:

- tree_root_expression.

The tree_root_expression specifies the root of the CSG tree.

See 4.3.276, 4.3.277, 4.3.278, 4.3.279, 4.3.280, and 4.3.281 for the application assertions.

4.2.101 Csg_primitive

A Csg_primitive is a basic solid object that may be combined with other solid objects using regularized Boolean operations in order to create more complex solid geometric elements.

A Csg_primitive is a type of Detailed_geometric_model_element (see 4.2.136).

Each Csg_primitive is either a Right_circular_cylinder (see 4.2.385), a Right_angular_wedge (see 4.2.383), a Right_circular_cone (see 4.2.384), a Torus (see 4.2.470), a Block (see 4.2.40), or a Sphere (see 4.2.420).

4.2.102 Curve

A Curve is a path of a point moving in its coordinate space.

A Curve is a type of Detailed_geometric_model_element (see 4.2.136).

Each Curve is either a Hyperbola (see 4.2.231), a Parabola (see 4.2.300), a Trimmed_curve (see 4.2.478), an Offset_curve (see 4.2.289), an Ellipse (see 4.2.159), a Composite_curve (see 4.2.79), a Circle (see 4.2.55), a Line (see 4.2.265), a B_spline_curve (see 4.2.36), a Curve_on_surface (see 4.2.104), or a Polyline (see 4.2.331).

4.2.103 Curve_appearance

A Curve_appearance governs the visual appearance of geometric curves and annotation curves.

The data associated with a Curve_appearance are the following:

- colour;

- font;
- width.

4.2.103.1 colour

The colour specifies the colour of the visible segments of a curve. The colour is either userdefined or predefined. Figure 19 shows the predefined colours.

4.2.103.2 font

The font specifies the line font of a curve. The font determines the visible and invisible segments of a curve. Figure 20 shows the predefined curve fonts.

4.2.103.3 width

The width specifies the thickness of the visible segments of a curve measured perpendicular to the direction of the curve. The width is invariant to the scale of a Model_image (see 4.2.284).

Colour name	Red	Green	Blue
Black	0	0	0
Red	1.0	0	0
Green	0	1.0	0
Blue	0	0	1.0
Yellow	1.0	1.0	0
Magenta	1.0	0	1.0
Cyan	0	1.0	1.0
White	1.0	1.0	1.0

Figure 19 – Predefined colours

4.2.104 Curve_on_surface

A Curve_on_surface is a curve that lies on a parametric surface. This includes the cases of a 3D curve representation together with the basis surface, a 2D curve representation within the parameter space of the surface, and a combination of both.

A Curve_on_surface is a type of Curve (see 4.2.102).

Curve pattern name	Segment (mm)	Space (mm)	Segment (mm)	Space (mm)	Segment (mm)	Space (mm)	Number of Segments
Continuous							0
Dotted	1.0	1.0					2
Dashed	4.0	1.5					2
Chain	7.0	1.0	1.0	1.0			4
Chain double dash	7.0	1.0	1.0	1.0	1.0	1.0	6

Figure 20 – Predefined curve fonts

4.2.105 Curve_set_2d

A Curve_set_2d is a collection of a two-dimensional Curve (see 4.2.102) or Point (see 4.2.322) objects that establish a Wireframe_model_2d (see 4.2.497).

A Curve_set_2d is a type of Detailed_geometric_model_element (see 4.2.136).

The data associated with a Curve_set_2d are the following:

– element.

The element specifies the set of Curve (see 4.2.102) or Point (see 4.2.322) objects that form a Curve_set_2d.

See 4.3.282 and 4.3.283 for the application assertions.

There shall be at least one object that defines the element for a Curve_set_2d.

4.2.106 Curve_set_3d

A Curve_set_3d is a collection of three-dimensional Curve (see 4.2.102) or Point (see 4.2.322) objects that establish a Wireframe_model_3d (see 4.2.498).

A Curve_set_3d is a type of Detailed_geometric_model_element (see 4.2.136).

The data associated with a Curve_set_3d are the following:

– element.

The element specifies the set of Curve (see 4.2.102) or Point (see 4.2.322) objects that form a Curve_set_3d.

See 4.3.284 and 4.3.285 for the application assertions.

There shall be at least one object that defines the element for a Curve_set_3d.

4.2.107 Curved_distance_dimension

A Curved_distance_dimension is the chordal dimension or the arc length along a curve.

A Curved_distance_dimension is a type of Location_dimension (see 4.2.270).

The data associated with a Curved_distance_dimension are the following:

- path;
- with_curve_direction.

4.2.107.1 path

The path specifies the Shape_aspect (see 4.2.394) representing the curve.

See 4.3.286 for the application assertion.

4.2.107.2 with_curve_direction

The with_curve_direction specifies a Boolean value that indicates the direction along the element to apply the dimension. A value of 'true' indicates that the dimension is applied from the start point of the curve to the end point of the curve. A value of 'false' indicates that the direction does not matter.

4.2.108 Cylindrical_surface

A Cylindrical_surface is a surface at a constant distance from a line. A Cylindrical_surface shall be trimmed to a finite extension if used in a Surface_model (see 4.2.439) or a Geometrically_-bounded_surface_model (see 4.2.220).

A Cylindrical_surface is a type of Surface (see 4.2.434).

4.2.109 Cylindricity_tolerance

A Cylindricity_tolerance is the limitation of the amount of deviation points of a cylindrical surface may have from being equidistant from the cylinder axis. It defines a tolerance zone bounded by two concentric cylinders within which the surface must lie.

A Cylindricity_tolerance is a type of Geometric_tolerance (see 4.2.216).

4.2.110 Data_environment

A Data_environment is the specification of the conditions under which a Material_property_-representation (see 4.2.280) is valid.

EXAMPLE 65 – A Data_environment may have the name 'standard' and the description '20 degrees Celsius, 75% humidity'.

The data associated with a Data_environment are the following:

- description;
- name.

4.2.110.1 description

The description specifies additional information about the Data_environment.

See 4.3.287 for the application assertion.

There shall be exactly one object that defines the description for a Data_environment.

4.2.110.2 name

The name specifies the word or group of words by which the Data_environment is referred to.

4.2.111 Date_and_person_assignment

A Date_and_person_assignment is an object that associates a Date_and_person_or_organization (see 4.2.112) with an object.

The data associated with a Date_and_person_assignment are the following:

- assigned_date_and_person;
- is_applied_to;
- role.

4.2.111.1 assigned_date_and_person

The assigned_date_and_person specifies the Date_and_person_or_organization (see 4.2.112).

See 4.3.297 for the application assertion.

4.2.111.2 is_applied_to

The is_applied_to specifies the object with which the Date_and_person_assignment is associated.

See 4.3.288, 4.3.289, 4.3.290, 4.3.291, 4.3.292, 4.3.293, 4.3.294, 4.3.295, 4.3.296, 4.3.298, 4.3.299, 4.3.300, 4.3.301, 4.3.302, 4.3.303, 4.3.304, 4.3.305, 4.3.306, 4.3.307, 4.3.308, 4.3.309, 4.3.310, 4.3.311, 4.3.312, 4.3.313, 4.3.314, 4.3.315, 4.3.316, 4.3.317, 4.3.318, 4.3.319, 4.3.320, 4.3.321, 4.3.322, 4.3.323, 4.3.324, 4.3.325, 4.3.326, and 4.3.327 for the application assertions.

There shall be exactly one object that the Date_and_person_assignment is_applied_to.

4.2.111.3 role

The role specifies the relationship between the date/time and the person or organization in the Date_and_person_assignment. Where applicable the following values shall be used:

- 'creation': The assignment specifies that the referenced object has been created by the given person or organization at the given date and time;

– 'update': The assignment specifies that the referenced object has been altered by the given person or organization at the given date and time.

4.2.112 Date_and_person_or_organization

A Date_and_person_or_organization is a Date_time (see 4.2.113) and a Person (see 4.2.307) or an Organization (see 4.2.294).

The data associated with a Date_and_person_or_organization are the following:

– actual_date;

– person_or_organization.

4.2.112.1 actual_date

The actual_date specifies the date component of a Date_and_person_or_organization.

See 4.3.328 for the application assertion.

4.2.112.2 person_or_organization

The person_or_organization specifies the Organization (see 4.2.294), the Person (see 4.2.307), or the Person_in_organization (see 4.2.308) that is part of the Date_and_person_or_organization.

See 4.3.329, 4.3.330, and 4.3.331 for the application assertions.

There shall be exactly one object that defines the person_or_organization for a Date_and_person_-or_organization.

4.2.113 Date_time

A Date_time is the specification of a date and an optional time of day.

The data associated with a Date_time are the following:

– date;

– time.

4.2.113.1 date

The date specifies the calendar time conveying information about the year, the month, and the day in no specific order.

4.2.113.2 time

The time specifies a moment of occurrence measured by hour, minute, and second.

The time need not be specified for a particular Date_time.

4.2.114 Date_time_assignment

A Date_time_assignment is an association of Date_time (see 4.2.113) with some product data.

The data associated with a Date_time_assignment are the following:

- assigned_date_time;
- is_applied_to;
- role.

4.2.114.1 assigned_date_time

The assigned_date_time specifies the Date_time (see 4.2.113) that is associated with the product data.

See 4.3.341 for the application assertion.

4.2.114.2 is_applied_to

The is_applied_to specifies the object of product data with which the Date_time_assignment is associated. There shall be exactly one object to which the Date_time_assignment is applied to.

See 4.3.332, 4.3.333, 4.3.334, 4.3.335, 4.3.336, 4.3.337, 4.3.338, 4.3.339, 4.3.340, 4.3.342, 4.3.343, 4.3.344, 4.3.345, 4.3.346, 4.3.347, 4.3.348, 4.3.349, 4.3.350, 4.3.351, 4.3.352, 4.3.353, 4.3.354, 4.3.355, 4.3.356, 4.3.357, 4.3.358, 4.3.359, 4.3.360, 4.3.361, 4.3.362, 4.3.363, 4.3.364, 4.3.365, 4.3.366, 4.3.367, 4.3.368, 4.3.369, 4.3.370, and 4.3.371 for the application assertions.

There shall be exactly one object that the Date_time_assignment is_applied_to.

4.2.114.3 role

The role specifies the action associated with the Date_time_assignment. Where applicable the following values shall be used:

- 'creation': The assignment specifies that the referenced object was created at the given date and time;
- 'update': The assignment specifies that the referenced object was altered at the given date and time;
- 'production': The assignment specifies that the referenced object (e.g. a physical instance) was produced at a given date and time;
- 'installation': The assignment specifies that the referenced object (e.g. a physical instnace) was mounted in a product at a given date and time;
- 'registration': The assignment specifies that the referenced object (e.g. a property) was determined at a given date and time.

4.2.115 Dated_configuration

A Dated_configuration is a Configuration (see 4.2.89) that applies from a given date on.

A Dated_configuration is a type of Configuration (see 4.2.89).

4.2.116 Datum

A Datum is a theoretically exact point, line, axis, or plane which indicates the origin of a specified dimensional relationship between a toleranced element and a designated element on a part. A designated part element serves as the Datum_feature (see 4.2.117), whereas, its true counterpart establishes the datum plane or axis. A Datum may be controlled by a Material_condition (see 4.2.278). The concept of datums for tolerancing is defined in ISO 5459.

Each Datum is either a Datum_feature (see 4.2.117) or a Compound_datum (see 4.2.80).

The data associated with a Datum are the following:

- precedence.

The precedence the number of the Datum in a sequence of Datum objects.

4.2.117 Datum_feature

A Datum_feature is an actual part element which contacts, or is used to establish, a Datum (see 4.2.116).

A Datum_feature is a type of Datum (see 4.2.116).

The data associated with a Datum_feature are the following:

- is_controlled_by;
- is_defined_by;
- name.

4.2.117.1 is_controlled_by

The is_controlled_by specifies the Material_condition (see 4.2.278) that is associated with the application of the Datum_feature.

See 4.3.374 for the application assertion.

The is_controlled_by need not be specified for a particular Datum_feature.

4.2.117.2 is_defined_by

The is_defined_by specifies the Datum_target_set (see 4.2.119), the Derived_geometry (see 4.2.128), or the Shape_aspect (see 4.2.394) that represents the Datum_feature.

See 4.3.372, 4.3.373, and 4.3.375 for the application assertions.

There shall be exactly one object that the Datum_feature is_defined_by.

4.2.117.3 name

The name specifies the word or group of words by which the Datum_feature is referred to.

4.2.118 Datum_target

A Datum_target is a designated geometric element on the surface of a part used to locate the part in a datum reference frame.

Each Datum_target is either a Target_area (see 4.2.446) or a Placed_target (see 4.2.314).

The data associated with a Datum_target are the following:

- id.

The id specifies the identifier of the Datum_target.

4.2.119 Datum_target_set

A Datum_target_set is a set of Datum_target (see 4.2.118) objects which is used to define a datum reference frame for a part.

The data associated with a Datum_target_set are the following:

- rule_description;
- target.

4.2.119.1 rule_description

The rule_description specifies the type of datum that is formed by the Datum_target_set.

EXAMPLE 66 – The rule_description 'V-block' indicates that two Datum_target (see 4.2.118) objects on a cylindrical element are supposed to form the areas of contact in a V-shaped fixture.

See 4.3.377 for the application assertion.

The rule_description need not be specified for a particular Datum_target_set.

If present, there shall be exactly one object that defines the rule_description for a Datum_target_set.

4.2.119.2 target

The target specifies the set of Datum_target (see 4.2.118) objects that form the Datum_target_set.

See 4.3.376 for the application assertion.

4.2.120 Default_chamfer

A Default_chamfer is the definition of a chamfer that shall be applied to each non-tangent continuous, convex edge of a shape model that has no other chamfer or edge round specified. For a given context only one of Default_chamfer and Default_edge_round (see 4.2.122) shall be specified at a time. A Default_chamfer is applicable only to edges that do not change from convex to concave.

EXAMPLE 67 – During the creation of a 3D CAD model the explicit modelling of chamfers is often omitted to save on time and model size.

The data associated with a Default_chamfer are the following:

- description;
- offset_value.

4.2.120.1 description

The description specifies additional information about the Default_chamfer.

See 4.3.379 for the application assertion.

The description need not be specified for a particular Default_chamfer.

If present, there shall be exactly one object that defines the description for a Default_chamfer.

4.2.120.2 offset_value

The offset_value specifies the width of the chamfer with respect to the two faces where it is applied to.

See 4.3.378 for the application assertion.

4.2.121 Default_draft_angle

A Default_draft_angle is the definition of an angle that is applied to each surface of the part that corresponds to the draft direction.

NOTE 1 – For parts manufactured by casting, the draft angle is necessary to facilitate the extraction of the part out of the form after the casting.

NOTE 2 – There may be various draft directions depending on the actual shape of the part. The Default_draft_angle is therefore not necessarily meant to enable a system to create a resulting shape out of the application of the angle; in most cases this is additional information that has to be interpreted by human beings.

The data associated with a Default_draft_angle are the following:

- description;
- draft_angle_value.

4.2.121.1 description

The description specifies additional information about the Default_draft_angle.

See 4.3.381 for the application assertion.

The description need not be specified for a particular Default_draft_angle.

If present, there shall be exactly one object that defines the description for a Default_draft_angle.

4.2.121.2 draft_angle_value

The draft_angle_value specifies the value of the Default_draft_angle.

See 4.3.380 for the application assertion.

4.2.122 Default_edge_round

A Default_edge_round is the definition of an edge round that shall be applied to non-tangent continuous, convex edge of a shape model that has no other edge round or chamfer specified. For a given context only one of Default_edge_round and Default_chamfer (see 4.2.120) shall be specified at a time. A Default_edge_round is applicable only to edges that do not change from convex to concave.

EXAMPLE 68 – During the creation of a 3D CAD model the explicit modelling of edge rounds is often omitted to save on time and model size.

The data associated with a Default_edge_round are the following:

- description;
- radius_value.

4.2.122.1 description

The description specifies additional information about the Default_edge_round.

See 4.3.383 for the application assertion.

The description need not be specified for a particular Default_edge_round.

If present, there shall be exactly one object that defines the description for a Default_edge_round.

4.2.122.2 radius_value

The radius_value specifies the amount of curvature for a convex transition between two faces.

See 4.3.382 for the application assertion.

4.2.123 Default_fillet

A Default_fillet is the definition of a fillet that shall be applied to each non-tangent continuous, concave edge of a shape model that has no other fillet specified and that does not change its nature from concave to convex.

EXAMPLE 69 – During the creation of a 3D CAD model the explicit modelling of fillets is often omitted to save on time and model size.

The data associated with a Default_fillet are the following:

- description;
- radius_value.

4.2.123.1 description

The description specifies additional information about the Default_fillet.

See 4.3.385 for the application assertion.

The description need not be specified for a particular Default_fillet.

If present, there shall be exactly one object that defines the description for a Default_fillet.

4.2.123.2 radius_value

The radius_value specifies the amount of curvature for a concave transition between two faces.

See 4.3.384 for the application assertion.

4.2.124 Default_roughness

A Default_roughness is the definition of a roughness that shall be applied to each surface of a part that has no other roughness specified.

The data associated with a Default_roughness are the following:

- description;
- max_value;
- min_value.

4.2.124.1 description

The description specifies additional information about the Default_roughness.

See 4.3.386 for the application assertion.

The description need not be specified for a particular Default_roughness.

If present, there shall be exactly one object that defines the description for a Default_roughness.

4.2.124.2 max_value

The max_value specifies the maximum roughness value that shall apply as default.

See 4.3.387 for the application assertion.

4.2.124.3 min_value

The min_value specifies the minimum roughness value that shall apply as default.

See 4.3.387 for the application assertion.

The min_value need not be specified for a particular Default_roughness.

4.2.125 Default_setting_association

A Default_setting_association is the setting of default values for one or more particular contexts.

The data associated with a Default_setting_association are the following:

- default_context;
- default_value.

4.2.125.1 default_context

The default_context specifies the Drawing (see 4.2.149), the External_document (see 4.2.162), the Item_shape (see 4.2.245), or the View_area (see 4.2.491) in whose context the default settings apply.

See 4.3.394, 4.3.395, 4.3.397, and 4.3.398 for the application assertions.

There shall be exactly one object that defines the default_context for a Default_setting_association.

4.2.125.2 default_value

The default_value specifies defaults of various kinds. The quantity and units of the default values are defined by the particular case of default.

See 4.3.388, 4.3.389, 4.3.390, 4.3.391, 4.3.392, 4.3.393, and 4.3.396 for the application assertions.

There shall be at least one object that defines the default_value for a Default_setting_association.

4.2.126 Default_thickness

A Default_thickness is the value of thickness that is valid for all sheet like objects in the context defined by the Default_setting_association (see 4.2.125) if no thickness is assigned explicitly.

The data associated with a Default_thickness are the following:

- description;

– thickness_value.

4.2.126.1 description

The description specifies additional information about the Default_thickness.

See 4.3.399 for the application assertion.

The description need not be specified for a particular Default_thickness.

If present, there shall be exactly one object that defines the description for a Default_thickness.

4.2.126.2 thickness_value

The thickness_value specifies the actual value with the corresponding unit for the Default_-thickness. The thickness_value is measured normal to the surface.

See 4.3.400 for the application assertion.

4.2.127 Defined_thread

A Defined_thread is a thread where the shape of the thread is specified by the values of the attributes. A Defined_thread may be an internal or an external thread.

EXAMPLE 70 – Figure 21 depicts a Defined_thread.

A Defined_thread is a type of Thread_specification (see 4.2.458).

The data associated with a Defined_thread are the following:

– crest;

– density;

– fit_class;

– form;

– hand;

– major_diameter;

– minor_diameter;

– pitch_diameter.

4.2.127.1 crest

The crest specifies the distance between the opposing points of the thread. The crest is formed by the intersection of the sides of the thread if extended, if necessary, beyond the top of the thread.

See 4.3.401 for the application assertion.

4.2.127.2 density

The density specifies the number of threads per the axial unit of length of measure.

See 4.3.401 for the application assertion.

4.2.127.3 fit_class

The fit_class specifies the class of fit specification for the Defined_thread as defined in ISO 965/1. These classes are distinguished from each other by the amount of tolerance and allowance.

EXAMPLE 71 – Examples for fit class are '4e' or '2A'.

4.2.127.4 form

The form specifies one of the standard thread forms. Various forms of threads are used to hold parts together, to adjust parts with reference to each other, or to transmit power.

EXAMPLE 72 – Examples of form are 'metric', 'square', 'unified', 'sharp v', 'buttress', 'standard worm', and 'knuckle'.

4.2.127.5 hand

The hand whether the thread is right or left handed. When viewed toward an end, a right hand winds in a clockwise direction and a left hand winds in a counterclockwise direction.

4.2.127.6 major_diameter

The major_diameter specifies the dimension of the largest diameter of a Defined_thread.

NOTE – The major_diameter applies to both an internal and an external thread.

See 4.3.401 for the application assertion.

4.2.127.7 minor_diameter

The minor_diameter specifies the dimension of the smallest diameter of a Defined_thread.

NOTE – The minor_diameter applies to both an internal and an external thread.

See 4.3.401 for the application assertion.

4.2.127.8 pitch_diameter

The pitch_diameter specifies the dimension of an imaginary cylinder passing through the threads so as to make equal the widths of the threads and the widths of the spaces cut by the cylinder.

See 4.3.401 for the application assertion.

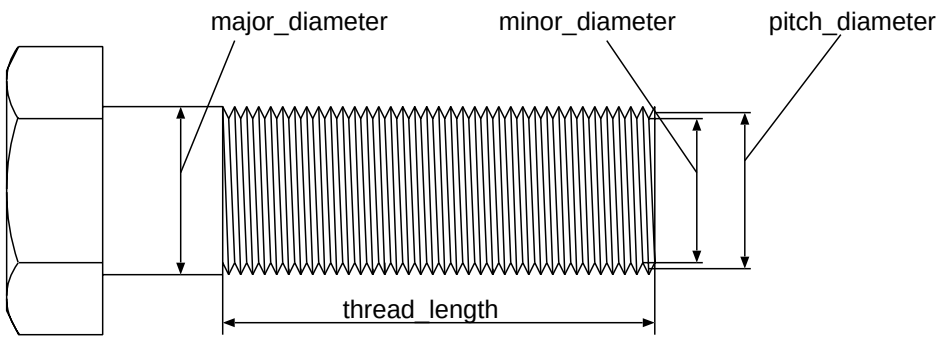


Figure 21 – Defined_thread

4.2.128 Derived_geometry

A Derived_geometry is the identification of a Constructive_geometric_element (see 4.2.94) that is derived from an existing aspect of the shape of a part for tolerancing purposes.

The data associated with a Derived_geometry are the following:

- geometry;
- is_derived_from;
- role.

4.2.128.1 geometry

The geometry specifies the Constructive_geometric_element (see 4.2.94) that is used for tolerancing.

See 4.3.402 for the application assertion.

4.2.128.2 is_derived_from

The is_derived_from specifies the Shape_aspect (see 4.2.394) from which the geometric element is derived.

See 4.3.403 for the application assertion.

4.2.128.3 role

The role specifies the function of the Constructive_geometric_element (see 4.2.94) that is used as Derived_geometry.

EXAMPLE 73 – 'symmetry plane' is an example for the role.

4.2.129 Descriptive_specification

A Descriptive_specification is a text description of characteristics to distinguish a final part from the corresponding neutral part.

The data associated with a Descriptive_specification are the following:

- description.

The description specifies the Descriptive_specification.

EXAMPLE 74 – The description may be e.g., 'arctic white' as description of a car body colour, 'chrome plated' as description of an exhaust pipe, or 'standard cloth' as description of the seat cover.

See 4.3.404 for the application assertion.

There shall be exactly one object that defines the description for a Descriptive_specification.

4.2.130 Design_constraint

A Design_constraint is a geometry based requirement that influences the design of an Item_solution (see 4.2.246).

EXAMPLE 75 – Ergonomic information like 'hip point' or 'forward driver vision' as well as 'door opening control lines', 'shipping critical dimensions', or 'stone throw zone' are examples for Design_constraint.

Each Design_constraint may be a Design_constraint_version (see 4.2.133).

The data associated with a Design_constraint are the following:

- constraint_id;
- description;
- is_valid_for;
- name.

4.2.130.1 constraint_id

The constraint_id specifies the identifier of the Design_constraint.

4.2.130.2 description

The description specifies additional information about the Design_constraint.

See 4.3.405 for the application assertion.

The description need not be specified for a particular Design_constraint.

If present, there shall be exactly one object that defines the description for a Design_constraint.

4.2.130.3 is_valid_for

The is_valid_for specifies a Product_class (see 4.2.348) that is affected by the Design_constraint.

See 4.3.406 for the application assertion.

4.2.130.4 name

The name specifies the word or group of words by which the Design_constraint is referred to.

See 4.3.405 for the application assertion.

The name need not be specified for a particular Design_constraint.

If present, there shall be exactly one object that defines the name for a Design_constraint.

4.2.131 Design_constraint_association

A Design_constraint_association is a mechanism to associate a Design_constraint (see 4.2.130) with a Product_component (see 4.2.350) or with an Item_solution (see 4.2.246).

The data associated with a Design_constraint_association are the following:

- is_based_on;
- is_constraining;
- name.

4.2.131.1 is_based_on

The is_based_on specifies the Design_constraint (see 4.2.130) that represents the constraint.

See 4.3.407 for the application assertion.

4.2.131.2 is_constraining

The is_constraining specifies the object that is affected by the Design_constraint (see 4.2.130).

See 4.3.408 and 4.3.410 for the application assertions.

There shall be exactly one object that the Design_constraint_association is_constraining.

4.2.131.3 name

The name specifies the word or group of words by which the Design_constraint_association is referred to.

See 4.3.409 for the application assertion.

The name need not be specified for a particular Design_constraint_association.

If present, there shall be exactly one object that defines the name for a Design_constraint_-association.

4.2.132 Design_constraint_relationship

A Design_constraint_relationship is a relationship between two Design_constraint (see 4.2.130) objects.

The data associated with a Design_constraint_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.132.1 description

The description specifies additional information about the Design_constraint_relationship.

See 4.3.412 for the application assertion.

The description need not be specified for a particular Design_constraint_relationship.

If present, there shall be exactly one object that defines the description for a Design_constraint_-relationship.

4.2.132.2 related

The related specifies the second of the two Design_constraint (see 4.2.130) objects related by the Design_constraint_relationship.

See 4.3.411 for the application assertion.

4.2.132.3 relating

The relating specifies the first of the two Design_constraint (see 4.2.130) objects related by the Design_constraint_relationship.

See 4.3.411 for the application assertion.

4.2.132.4 relation_type

The relation_type specifies the meaning of the relationship.Where applicable the following values shall be used:

– 'decomposition': The application object defines a relationship where the related Design_constraint (see 4.2.130) is one of the components into which the relating Design_constraint is broken down;

– 'precedence': The application object defines a relationship where the related Design_constraint has higher priority than the relating Design_constraint.

4.2.133 Design_constraint_version

A Design_constraint_version is a particular version of a Design_constraint (see 4.2.130).

A Design_constraint_version is a type of Design_constraint (see 4.2.130).

The data associated with a Design_constraint_version are the following:

– version_id.

The version_id specifies the identification of a particular version of a Design_constraint (see 4.2.130).

4.2.134 Design_discipline_item_definition

A Design_discipline_item_definition is a view of an Item_version (see 4.2.248) relevant for the requirements of one or more life cycle stages and application domains. This view collects product data for a specific task.

NOTE – The selection of data describing an Item may be different for assembly purposes, packing purposes or analysis purposes.

Each Design_discipline_item_definition may be one of the following: a Mating_definition (see 4.2.282), a Collection_definition (see 4.2.77), an Assembly_definition (see 4.2.27), or a Process_state (see 4.2.347).

The data associated with a Design_discipline_item_definition are the following:

– associated_item_version;

– id;

– is_relevant_for;

– name.

4.2.134.1 associated_item_version

The associated_item_version specifies the Item_version (see 4.2.248) for which the Design_discipline_item_definition is a view.

See 4.3.414 for the application assertion.

4.2.134.2 id

The id specifies the identifier of the Design_discipline_item_definition.

4.2.134.3 is_relevant_for

The is_relevant_for specifies the Application_context (see 4.2.20) in which the Design_discipline_item_definition has to be considered.

See 4.3.413 for the application assertion.

4.2.134.4 name

The name specifies the word or group of words used to refer to the Design_discipline_item_definition.

See 4.3.415 for the application assertion.

The name need not be specified for a particular Design_discipline_item_definition.

If present, there shall be exactly one object that defines the name for a Design_discipline_item_definition.

4.2.135 Detailed_element

A Detailed_element is a single element of a Geometric_model (see 4.2.211) or of a Styled_model (see 4.2.429) or a kinematic element.

Each Detailed_element is either a Detailed_model_element (see 4.2.137) or a Styled_element (see 4.2.427).

The data associated with a Detailed_element are the following:

- name.

The name specifies the word or group of words by which the Detailed_element is referred to.

See 4.3.416 for the application assertion.

The name need not be specified for a particular Detailed_element.

If present, there shall be exactly one object that defines the name for a Detailed_element.

4.2.136 Detailed_geometric_model_element

A Detailed_geometric_model_element is a geometric element that may be used to define a Geometric_model (see 4.2.211).

A Detailed_geometric_model_element is a type of Detailed_model_element (see 4.2.137).

Each Detailed_geometric_model_element is either a Geometric_set_3d (see 4.2.215), a Topologically_connected_set (see 4.2.468), a Point_direction (see 4.2.324), a Boolean_result (see 4.2.41), a Point_direction_sequence (see 4.2.326), a Half_space_solid (see 4.2.224), a Curve_set_2d (see

4.2.105), a B_rep (see 4.2.34), a Faceted_b_rep (see 4.2.171), a Point_sequence (see 4.2.329), a Face (see 4.2.167), a Curve_set_3d (see 4.2.106), a Shell (see 4.2.398), a Curve (see 4.2.102), a Csg_primitive (see 4.2.101), a Surface (see 4.2.434), a Swept_face_solid (see 4.2.443), a Point (see 4.2.322), or a Solid_replica (see 4.2.411).

4.2.137 Detailed_model_element

A Detailed_model_element is a single geometric element of a geometric model.

A Detailed_model_element is a type of Detailed_element (see 4.2.135).

Each Detailed_model_element is either a Kinematic_link (see 4.2.255), a Template_instance (see 4.2.453), a Detailed_geometric_model_element (see 4.2.136), a Kinematic_structure (see 4.2.259), or a Kinematic_joint (see 4.2.254).

4.2.138 Diameter_taper

A Diameter_taper is a taper that has the grade of the taper specified by the final (smaller) diameter. A Diameter_taper shall only be applied to features that are defined by a swept Circular_closed_profile (see 4.2.57). The final diameter is measured at the end of the sweep path.

NOTE – Tapers are applied after general end conditions are applied.

EXAMPLE 76 – Figure 22 illustrates a Diameter_taper.

The data associated with a Diameter_taper are the following:

– final_diameter.

The final_diameter specifies the diameter of the circle at the end of the taper.

NOTE – The diameter at the beginning of the taper is given by the referencing feature itself.

See 4.3.417 for the application assertion.

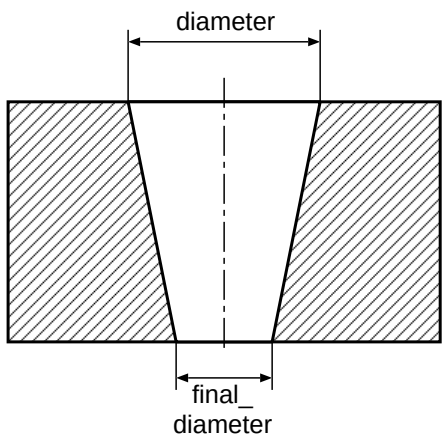


Figure 22 – Diameter_taper

4.2.139 Digital_document

A Digital_document is a piece of product data that is archived in a digital format.

A Digital_document is a type of External_document (see 4.2.162).

The data associated with a Digital_document are the following:

- application_domain;
- character_code;
- creating_interface;
- creating_system;
- file_name;
- operating_system;
- syntactical_format.

4.2.139.1 application_domain

The application_domain specifies the domain where the computer interpretable data belong to.

EXAMPLE 77 – 'numerical control', 'robotics control', 'measurement control', 'text', 'raster image', and 'vector image', 'audio' and 'video' are examples for application_domain.

4.2.139.2 character_code

The character_code specifies the character code used for the stored data.

EXAMPLE 78 – 'ASCII', 'ISO 646 IRV', 'ISO 8859-1', EBCDIC', or 'BINARY' are examples for character_code.

4.2.139.3 creating_interface

The creating_interface specifies the interface and the interface version that was used to create the data.

The creating_interface need not be specified for a particular Digital_document.

4.2.139.4 creating_system

The creating_system specifies the system and the system version that was used to create the data.

The creating_system need not be specified for a particular Digital_document.

4.2.139.5 file_name

The file_name specifies the information necessary to access the Digital_document.

NOTE – Usually this includes the complete path and file name.

The file_name need not be specified for a particular Digital_document.

4.2.139.6 operating_system

The operating_system specifies the operating system and the operating system version that was used to create the data.

The operating_system need not be specified for a particular Digital_document.

4.2.139.7 syntactical_format

The syntactical_format specifies the syntactical format used for the stored data.

NOTE – Syntactical formats may be proprietary, native formats, or defined in standardized documents.

4.2.140 Dimension

A Dimension is the graphical presentation of the dimension value, associated information, and the symbology necessary to accurately depict its area of application. This includes linear dimensions, angular dimensions, curve dimensions, diameter dimensions, ordinate dimensions, radius dimensions, and leader directed dimensions.

EXAMPLE 79 – Examples for the symbology are headers, projection curves, or dimension curves.

A Dimension is a type of Annotation_element (see 4.2.15).

The data associated with a Dimension are the following:

- composition_curve;
- id;
- primary_callout;
- secondary_callout.

4.2.140.1 composition_curve

The composition_curve specifies the annotation curves that compose the Dimension.

See 4.3.418 for the application assertion.

4.2.140.2 id

The id specifies the identifier of the Dimension.

4.2.140.3 primary_callout

The primary_callout specifies the dimension value, tolerance information, and any associated notes given in the primary unit of measure for the corresponding View (see 4.2.490), Drawing_sheet (see 4.2.152), or View_area (see 4.2.491).

See 4.3.419 for the application assertion.

4.2.140.4 secondary_callout

The secondary_callout specifies the dimension value, tolerance information, and any associated notes given in the secondary unit of measure for the corresponding View (see 4.2.490), Drawing_sheet (see 4.2.152), or View_area (see 4.2.491).

EXAMPLE 80 – The primary unit of measure used by primary_callout could be millimeters and the secondary unit of measure used by secondary_callout could be inches.

See 4.3.419 for the application assertion.

The secondary_callout need not be specified for a particular Dimension.

4.2.141 Dimension_callout

A Dimension_callout is the text and symbols in the presentation of a dimension that represent the dimension value, dimension units, tolerance information and any related notes.

The data associated with a Dimension_callout are the following:

- is_displaying.

The is_displaying specifies the Geometric_dimension (see 4.2.210) that is displayed by the Dimension_callout.

See 4.3.420 for the application assertion.

4.2.142 Dimension_curve

A Dimension_curve is used for the graphical representation of a dimension along with other symbology, if necessary, to show the extent of the dimension.

EXAMPLE 81 – Figure 23 shows the use of a Dimension_curve on a drawing.

A Dimension_curve is a type of Annotation_curve (see 4.2.14).

The data associated with a Dimension_curve are the following:

- terminator.

The terminator specifies the terminating symbol for a particular Dimension_curve.

See 4.3.421 for the application assertion.

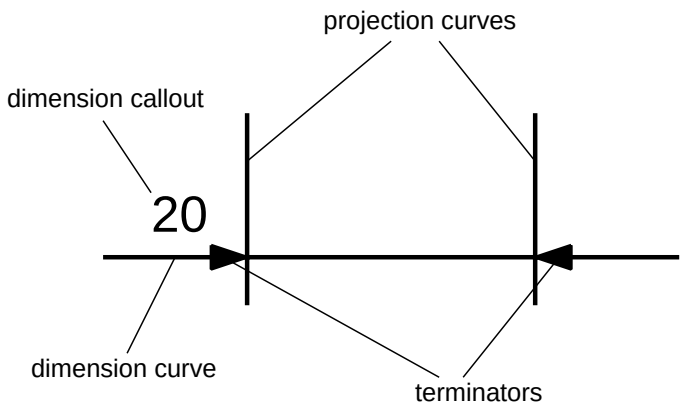


Figure 23 – Components of a dimension on a drawing

4.2.143 Dimension_sequence_pair

A Dimension_sequence_pair is the relationship between two adjacent dimensions that share a projection curve. This may be a chained or a parallel dimension pair.

The data associated with a Dimension_sequence_pair are the following:

- predecessor;
- successor.

4.2.143.1 predecessor

The predecessor specifies the draughting dimension that plays the role of a predecessor within a particular Dimension_sequence_pair (see 4.2.143).

See 4.3.422 for the application assertion.

4.2.143.2 successor

The successor specifies the draughting dimension that plays the role of a successor within a particular Dimension_sequence_pair (see 4.2.143).

See 4.3.422 for the application assertion.

4.2.144 Dimension_tolerance

A Dimension_tolerance is a mechanism to associate a Geometric_dimension (see 4.2.210) with a Limits_and_fits (see 4.2.264), a Plus_minus_tolerance (see 4.2.318), a Value_limit (see 4.2.482), or with a Value_range (see 4.2.483) or to associate a Feature_parameter (see 4.2.177) with a Limits_-and_fits or with a Plus_minus_tolerance. The Dimension_tolerance shall not be used to associate

a Feature_parameter with a Value_limit or with a Value_range. Besides the Dimension_tolerance adds information to the association that further specifies the application of the tolerance.

The data associated with a Dimension_tolerance are the following:

- envelope;
- is_defined_by;
- is_tolerancing;
- name;
- significant_digits;
- value_interpretation.

4.2.144.1 envelope

The envelope specifies whether each geometric constraint has to be fulfilled in itself or not. The envelope of the perfect shape corresponding to the maximum material shall not be larger than the specified dimension and tolerance. The envelope concept is defined in ISO 8015.

4.2.144.2 is_defined_by

The is_defined_by specifies the Limits_and_fits (see 4.2.264), the Plus_minus_tolerance (see 4.2.318), the Value_limit (see 4.2.482), or the Value_range (see 4.2.483) that is associated with the Feature_parameter (see 4.2.177) or with the Geometric_dimension (see 4.2.210).

See 4.3.425, 4.3.427, 4.3.428, and 4.3.429 for the application assertions.

There shall be exactly one object that the Dimension_tolerance is_defined_by.

4.2.144.3 is_tolerancing

The is_tolerancing specifies the Feature_parameter (see 4.2.177) or the Geometric_dimension (see 4.2.210) that is constrained by a tolerance.

See 4.3.423 and 4.3.424 for the application assertions.

There shall be exactly one object that the Dimension_tolerance is_tolerancing.

4.2.144.4 name

The name specifies the word or group of words by which the Dimension_tolerance is referred to.

See 4.3.426 for the application assertion.

The name need not be specified for a particular Dimension_tolerance.

If present, there shall be exactly one object that defines the name for a Dimension_tolerance.

4.2.144.5 significant_digits

The significant_digits specifies the number of decimal places indicating the required accuracy of the tolerance.

The significant_digits need not be specified for a particular Dimension_tolerance.

4.2.144.6 value_interpretation

The value_interpretation specifies information on how the Dimension_tolerance shall be interpreted.

NOTE – A Dimension_tolerance may be specified in the design stage of a product but it may also be documented as measured on a prototype.

EXAMPLE 82 – The value_interpretation may be e.g., 'required', 'designed', 'estimated', or 'measured'.

4.2.145 Directed_taper

A Directed_taper is a taper applied for drafting reasons that has the grade of the taper specified by an angle. The angle is measured from the specified draft_direction.

NOTE – Tapers are applied after general end conditions are applied.

The data associated with a Directed_taper are the following:

- draft_angle;
- draft_direction.

4.2.145.1 draft_angle

The draft_angle specifies the Feature_parameter (see 4.2.177) that represents the draft_angle. An angle between 0 and 90 degrees or between -90 and -180 degrees indicates that the profile grows larger along the draft direction. An angle between 0 and -90 degrees or between 90 and 180 degrees indicates that the profile grows smaller along the draft direction.

See 4.3.430 for the application assertion.

4.2.145.2 draft_direction

The draft_direction specifies the reference direction from which the taper angle is measured. The draft_direction is defined in the same Cartesian_coordinate_space (see 4.2.44) as the concerned Process_feature_in_solid (see 4.2.335).

4.2.146 Document_relationship

A Document_relationship is a relationship between two External_document (see 4.2.162) objects.

The data associated with a Document_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.146.1 description

The description specifies additional information about the Document_relationship.

See 4.3.432 for the application assertion.

The description need not be specified for a particular Document_relationship.

If present, there shall be exactly one object that defines the description for a Document_relationship.

4.2.146.2 related

The related specifies the second of the two External_document (see 4.2.162) objects related by the Document_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.431 for the application assertion.

4.2.146.3 relating

The relating specifies the first of the two External_document (see 4.2.162) objects related by the Document_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.431 for the application assertion.

4.2.146.4 relation_type

The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:

- 'copy': The application object defines a relationship where the related External_document (see 4.2.162) is a copy of the relating External_document;

- 'derivation': The application object defines a relationship where the related External_document is derived from the relating External_document (e.g. a pixel image may be derived from 3D CAD model);

– 'decomposition': The application object defines a relationship where the related External_document is one of the sub documents of the relating External_document (e.g. a decomposition of a document into clauses or into several CAD model files as a digital document;

– 'sequence': The application object defines a logical sequence where the related External_document comes after the relating External_document (e.g. sequence of clauses);

– 'substitution': The application object defines a relationship where the related External_document replaces the relating External_document;

– 'version association': The application object defines where the related External_document is a version of the relating External_document. In this case the related External_document shall be an External_document_version (see 4.2.164);

– 'version sequence': The application object defines a version sequence where the relating External_document is the preceding version and the related External_document (see 4.2.162) is the following version. In this case both External_document objects shall be of type External_document_version.

4.2.147 Document_type

A Document_type is the representation of the kind of External_document (see 4.2.162) objects or of Drawing (see 4.2.149) objects.

EXAMPLE 83 – 'requirement specification', 'test report', 'CAD model', 'technical drawing', or 'legal regulation' are examples for Document_type.

The data associated with a Document_type are the following:

– id;

– used_classification_system.

4.2.147.1 id

The id specifies the identifier of the Document_type.

4.2.147.2 used_classification_system

The used_classification_system specifies the Classification_system (see 4.2.70) that contains the information about how to interpret the identifier of the Document_type.

See 4.3.433 for the application assertion.

The used_classification_system need not be specified for a particular Document_type.

4.2.148 Draughting_callout

A Draughting_callout is a combination of text, annotation curves, and symbology that conveys information about a specific element or area.

EXAMPLE 84 – Figure 24 shows typical Draughting_callout objects on a drawing.

A Draughting_callout is a type of Annotation_element (see 4.2.15).

4.2.149 Drawing

A Drawing is the presentation of product data in a human interpretable form wherein the physical and functional requirements for that product are presented by pictures or text.

The data associated with a Drawing are the following:

- drawing_id;
- drawing_revision_id;
- drawing_specification;
- drawing_type;
- language;
- title.

4.2.149.1 drawing_id

The drawing_id specifies the identifier of a particular Drawing. The drawing_id shall be unique within the scope of the referenced item.

4.2.149.2 drawing_revision_id

The drawing_revision_id specifies the identifier of the revision of the drawing.

4.2.149.3 drawing_specification

The drawing_specification specifies the identification of a standard that the drawing conforms to. This standard specifies the presentation forms used in the drawing.

There may be more than zero drawing_specification for a Drawing.

4.2.149.4 drawing_type

The drawing_type specifies the category of the Drawing (see 4.2.149) and may indicate the information content.

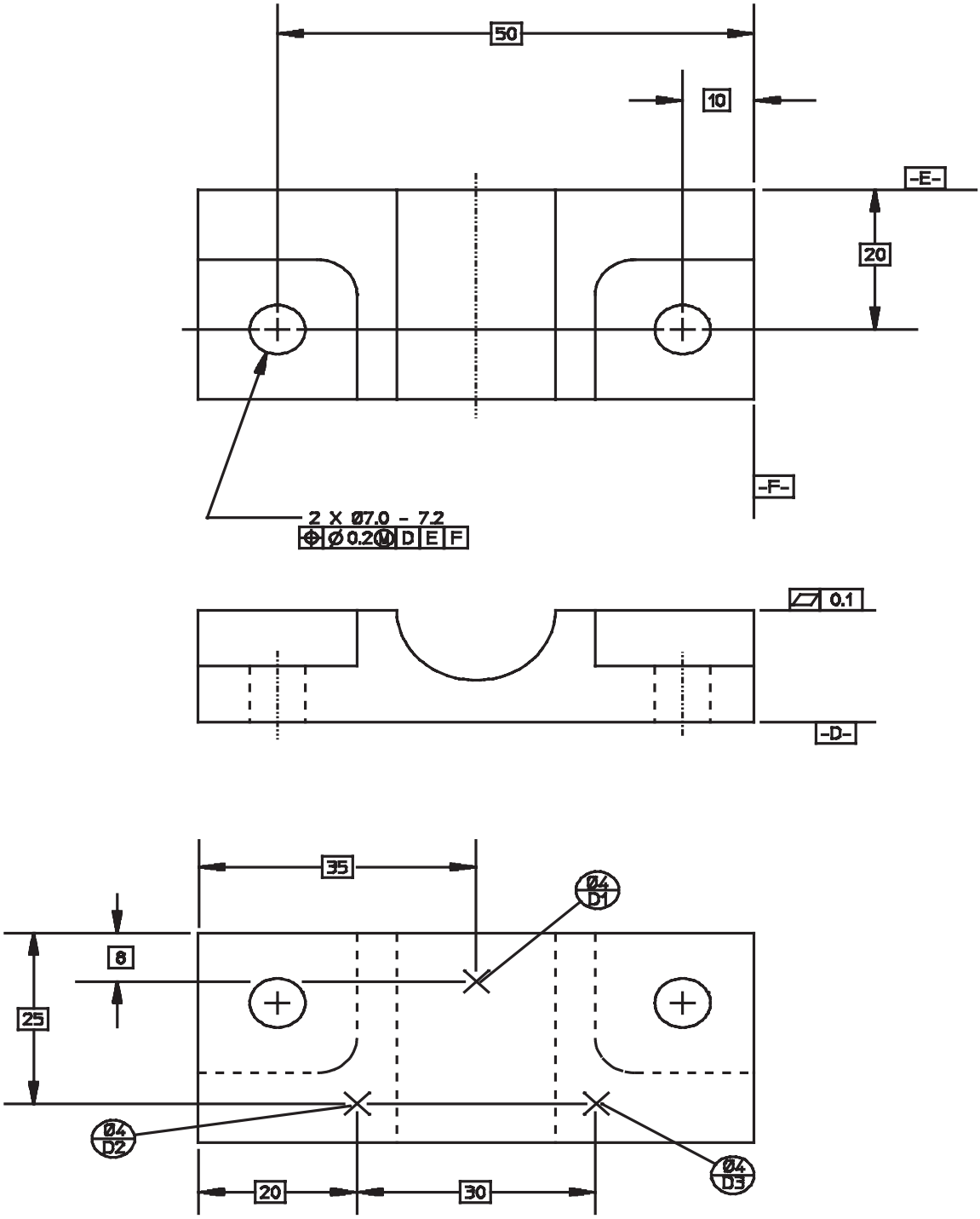


Figure 24 – Draughting callouts on a drawing

See 4.3.434 for the application assertion.

4.2.149.5 language

The language specifies the language that is used for text information on the Drawing.

There may be more than zero language for a Drawing.

4.2.149.6 title

The title specifies the description of the content or subject matter of the Drawing (see 4.2.149).

EXAMPLE 85 – 'fender' and 'assembly of engine' are examples for title.

See 4.3.435 for the application assertion.

The title need not be specified for a particular Drawing.

If present, there shall be exactly one object that defines the title for a Drawing.

4.2.150 Drawing_assignment

A Drawing_assignment is a mechanism to associate a Drawing (see 4.2.149) with the object that is presented on the Drawing.

The data associated with a Drawing_assignment are the following:

- drawing;
- is_describing.

4.2.150.1 drawing

The drawing specifies the Drawing (see 4.2.149) that is associated with the object.

See 4.3.439 for the application assertion.

4.2.150.2 is_describing

The is_describing specifies the object that is associated with the Drawing (see 4.2.149).

See 4.3.436, 4.3.437, 4.3.438, 4.3.440, 4.3.441, 4.3.442, 4.3.443, 4.3.444, 4.3.445, 4.3.446, 4.3.447, 4.3.448, 4.3.449, 4.3.450, and 4.3.451 for the application assertions.

There shall be exactly one object that the Drawing_assignment is_describing.

4.2.151 Drawing_sequence

A Drawing_sequence is a revision sequence between two Drawing (see 4.2.149) objects.

The data associated with a Drawing_sequence are the following:

- predecessor;
- successor.

4.2.151.1 predecessor

The predecessor specifies the old revision of the Drawing (see 4.2.149).

See 4.3.452 for the application assertion.

4.2.151.2 successor

The successor specifies the new revision of the Drawing (see 4.2.149).

See 4.3.452 for the application assertion.

4.2.152 Drawing_sheet

A Drawing_sheet is a logical division of a drawing into a two-dimensional area for the presentation of product data. These divisions correspond to sheet paper sizes for plotting.

The data associated with a Drawing_sheet are the following:

- associated_drawing;
- sheet_id;
- sheet_revision_id;
- sheet_space;
- size;
- title.

4.2.152.1 associated_drawing

The associated_drawing specifies the Drawing (see 4.2.149) that is valid for the Drawing_sheet.

See 4.3.454 for the application assertion.

4.2.152.2 sheet_id

The sheet_id specifies the page number for a particular Drawing_sheet.

4.2.152.3 sheet_revision_id

The sheet_revision_id specifies the identifier of the sheet revision.

4.2.152.4 sheet_space

The sheet_space specifies the Cartesian_coordinate_space_2d (see 4.2.45) that is valid for the drawing sheet.

See 4.3.453 for the application assertion.

4.2.152.5 size

The size specifies the width and the height of the Drawing_sheet.

4.2.152.6 title

The title specifies a description of the content or subject matter of the Drawing_sheet.

See 4.3.455 for the application assertion.

The title need not be specified for a particular Drawing_sheet.

If present, there shall be exactly one object that defines the title for a Drawing_sheet.

4.2.153 Edge_loop

An Edge_loop is a closed loop made up of a sequence of edges. In this sequence the start vertex of an edge coincides with the end vertex of its predecessors edge and the first vertex of the Edge_loop coincides with the last vertex.

The data associated with an Edge_loop are the following:

- edge.

The edge specifies the edges that compose the Edge_loop.

See 4.3.456 for the application assertion.

4.2.154 Edge_round

An Edge_round is a convex circular arc transition between two intersecting Face (see 4.2.167) objects without any constraints concerning changes of the radius along the Edge_round.

An Edge_round is a type of Transition_feature (see 4.2.476).

Each Edge_round may be a Constant_radius_edge_round (see 4.2.92).

4.2.155 Edge_transition

An Edge_transition is a state of geometric continuity for the intersection of two adjacent edges.

EXAMPLE 86 – Figure 25 depicts an Edge_transition.

The data associated with an Edge_transition are the following:

- edge_1;

- edge_2;
- transition;
- vertex_joint.

4.2.155.1 edge_1

The edge_1 specifies the first of the two adjacent Oriented_edge_curve (see 4.2.297) objects associated with the Edge_transition.

See 4.3.457 for the application assertion.

4.2.155.2 edge_2

The edge_2 specifies the second of the two Oriented_edge_curve (see 4.2.297) objects associated with the Edge_transition.

See 4.3.457 for the application assertion.

4.2.155.3 transition

The transition specifies the state of geometric continuity from the end vertex of the first edge to the start vertex of the second edge by a text description.

The following values shall be used:

- 'g0': G_0 continuity between two curves is a condition stating that the connected segments have common end/start points [9];
- 'g1': G_1 continuity between two curves is a condition set on the direction of the two tangents in the connection point between the curves; for true G_1 continuity no angular difference is allowed [9];
- 'g2': G_2 continuity between two curves is a condition set on the radius of curvature of the two curves in the connection point; for true G_2 continuity the radii in the connection point must be the same and the circles of curvature lie in the same plane [9].

4.2.155.4 vertex_joint

The vertex_joint specifies the vertex at the intersection of the two edges the continuity is described for.

See 4.3.458 for the application assertion.

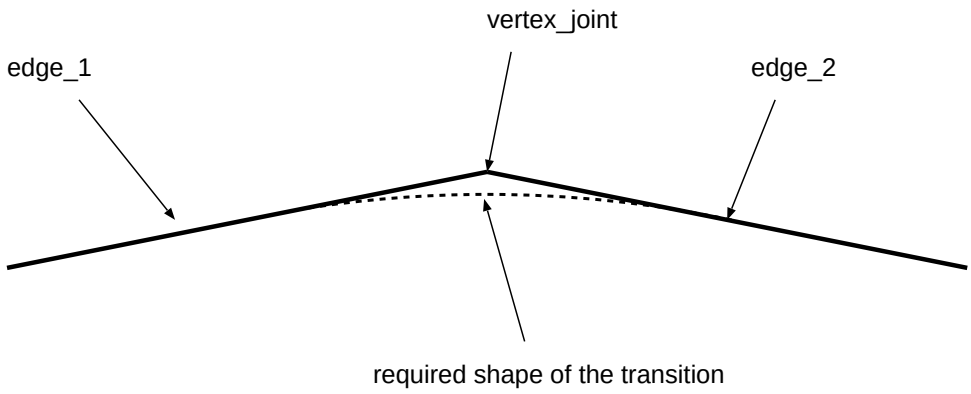


Figure 25 – Edge transition

4.2.156 Effectivity

An Effectivity is the identification of the valid use of an aspect of product data tracked by date. The Effectivity shall be specified either by one or both of the attributes 'start_definition' or 'end_definition' or by another 'master' Effectivity using an Effectivity_relationship (see 4.2.157).

The data associated with an Effectivity are the following:

- concerned_organization;
- description;
- effectivity_context;
- end_definition;
- is_applied_to;
- role;
- start_definition.

4.2.156.1 concerned_organization

The concerned_organization specifies the Organization (see 4.2.294) for which the Effectivity is valid.

EXAMPLE 87 – The Effectivity of the same item may be different in the various production sites of a car manufacturer.

See 4.3.481 for the application assertion.

The concerned_organization need not be specified for a particular Effectivity.

4.2.156.2 description

The description specifies additional information about the Effectivity.

See 4.3.480 for the application assertion.

The description need not be specified for a particular Effectivity.

If present, there shall be exactly one object that defines the description for an Effectivity.

4.2.156.3 effectivity_context

The effectivity_context specifies the life cycle stage for which the Effectivity is valid.

EXAMPLE 88 – The effectivity_context may be e.g., 'prototype building' in order to express that the Effectivity is valid for this life cycle stage of the concerned object.

The effectivity_context need not be specified for a particular Effectivity.

4.2.156.4 end_definition

The end_definition specifies the date or event or the period of time after the start_definition or after an event when the identified object is not effective any more.

EXAMPLE 89 – '15 years after end of production' is an example for an end_definition that can be expressed using an Event_reference (see 4.2.160) with offset.

See 4.3.465, 4.3.469, and 4.3.495 for the application assertions.

The end_definition need not be specified for a particular Effectivity.

If present, there shall be exactly one object that defines the end_definition for an Effectivity.

4.2.156.5 is_applied_to

The is_applied_to specifies the aspect of product data the Effectivity is associated with.

See 4.3.459, 4.3.460, 4.3.461, 4.3.462, 4.3.463, 4.3.464, 4.3.466, 4.3.467, 4.3.468, 4.3.470, 4.3.471, 4.3.472, 4.3.473, 4.3.474, 4.3.475, 4.3.476, 4.3.477, 4.3.478, 4.3.479, 4.3.482, 4.3.483, 4.3.484, 4.3.485, 4.3.486, 4.3.487, 4.3.488, 4.3.489, 4.3.490, 4.3.491, 4.3.492, 4.3.493, 4.3.494, and 4.3.496 for the application assertions.

There shall be at least one object that the Effectivity is_applied_to.

4.2.156.6 role

The role specifies the meaning of the Effectivity. Where applicable the following values shall be used:

- 'actual': The dates associated with the Effectivity are actual dates in the past;

- 'planned': The dates associated with the Effectivity are planned dates in the future concerning the point in time when the Effectivity is assigned;

- 'required': The dates associated with the Effectivity are due dates that have to be kept'.

4.2.156.7 start_definition

The start_definition specifies the date or the event when the identified object becomes effective.

See 4.3.465 and 4.3.469 for the application assertions.

The start_definition need not be specified for a particular Effectivity.

If present, there shall be exactly one object that defines the start_definition for an Effectivity.

4.2.157 Effectivity_relationship

An Effectivity_relationship is a relationship between two Effectivity (see 4.2.156) objects.

NOTE – Sometimes the effectivity is not dependent on particular dates but on the effectivity of other items. In this case the dates are not instantiated and there is an Effectivity_relationship to the reference Effectivity.

The data associated with an Effectivity_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.157.1 description

The description specifies additional information about the Effectivity_relationship.

See 4.3.498 for the application assertion.

The description need not be specified for a particular Effectivity_relationship.

If present, there shall be exactly one object that defines the description for an Effectivity_-relationship.

4.2.157.2 related

The related specifies the the second of the two Effectivity (see 4.2.156) objects related by the Effectivity_relationship.

NOTE – The semantic of this attribute is defined by the attribute 'relation_type'.

See 4.3.497 for the application assertion.

4.2.157.3 relating

The relating specifies the first of the two Effectivity (see 4.2.156) related by the Effectivity_-relationship.

NOTE – The semantic of this attribute is defined by the attribute 'relation_type'.

See 4.3.497 for the application assertion.

4.2.157.4 relation_type

The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:

- 'constraint': The time period between the start and end definition of the related Effectivity (see 4.2.156) shall be within the time period of the relating Effectivity;

- 'inheritance': The related Effectivity shall not have a 'start_definition' and 'end_definition' specified but inherits the effectivity dates from the relating Effectivity.

4.2.158 Element_delivery

An Element_delivery is the specification of the delivery of an Activity_element (see 4.2.3).

The data associated with an Element_delivery are the following:

- deliverable_element;

- destination;

- quantity.

4.2.158.1 deliverable_element

The deliverable_element specifies the Activity_element (see 4.2.3) that is subject to the delivery. This Activity_element shall be an Item_version (see 4.2.248), a Product_specification (see 4.2.354), or a Physical_instance (see 4.2.310).

See 4.3.499 for the application assertion.

4.2.158.2 destination

The destination specifies the Organization (see 4.2.294) the Activity_element (see 4.2.3) is to be delivered to.

See 4.3.500 for the application assertion.

4.2.158.3 quantity

The quantity specifies the number of Activity_element (see 4.2.3) objects to be delivered.

See 4.3.501 for the application assertion.

4.2.159 Ellipse

An Ellipse is a conic section defined by the lengths of the semi-major and semi-minor diameters and by the position (centre or mid point of the line joining the foci) and orientation of the curve.

An Ellipse is a type of Curve (see 4.2.102).

4.2.160 Event_reference

An Event_reference is the definition of a point in time established with respect to an event.

The data associated with an Event_reference are the following:

- event_context;
- event_type;
- offset.

4.2.160.1 event_context

The event_context specifies the piece of product data the Event_reference refers to.

EXAMPLE 90 – In the case of an Event_reference with event_type 'start of production' the event_-context would refer to the Item_version (see 4.2.248) that is subject to production.

See 4.3.502, 4.3.503, 4.3.504, 4.3.505, 4.3.506, 4.3.507, 4.3.508, 4.3.509, 4.3.510, 4.3.511, 4.3.512, 4.3.513, 4.3.514, 4.3.515, 4.3.516, 4.3.517, 4.3.518, 4.3.519, 4.3.520, 4.3.521, 4.3.522, 4.3.523, 4.3.524, 4.3.525, 4.3.526, 4.3.527, 4.3.528, 4.3.529, 4.3.530, 4.3.531, 4.3.532, 4.3.533, 4.3.534, 4.3.535, 4.3.536, 4.3.537, 4.3.538, 4.3.540, and 4.3.541 for the application assertions.

The event_context need not be specified for a particular Event_reference.

If present, there shall be exactly one object that defines the event_context for an Event_reference.

4.2.160.2 event_type

The event_type specifies the kind of event that serves as reference.

EXAMPLE 91 – 'start of production' or 'end of production' are examples for the event_type.

4.2.160.3 offset

The offset specifies the amount of time before or after the defined event that shall be used to calculate the actual Event_reference.

See 4.3.539 for the application assertion.

The offset need not be specified for a particular Event_reference.

4.2.161 Exclusive_process_operation

An Exclusive_process_operation is a mechanism to identify two or more Process_operation (see 4.2.336) objects that shall not have any overlap in time of execution.

The data associated with an Exclusive_process_operation are the following:

- element.

The element specifies the Process_operation (see 4.2.336) objects that shall not be executed at the same time.

See 4.3.542 for the application assertion.

4.2.162 External_document

An External_document is an externally defined reference that describes one or more objects.

Each External_document may be an External_document_version (see 4.2.164).

Each External_document may be one of the following: an Archived_document (see 4.2.24), a Digital_document (see 4.2.139), or a Physical_model (see 4.2.312).

The data associated with an External_document are the following:

- description;
- document_id;
- document_type;
- language;
- name.

4.2.162.1 description

The description specifies additional information about the External_document.

See 4.3.544 for the application assertion.

The description need not be specified for a particular External_document.

If present, there shall be exactly one object that defines the description for an External_-document.

4.2.162.2 document_id

The document_id specifies the identifier of the External_document.

4.2.162.3 document_type

The document_type specifies the kind of External_document.

See 4.3.543 for the application assertion.

The document_type need not be specified for a particular External_document.

4.2.162.4 language

The language specifies the language as defined in ISO 639–2 that is used for text information in the External_document.

There may be more than zero language for an External_document.

4.2.162.5 name

The name specifies the word or group of words by which the External_document is referred to.

See 4.3.544 for the application assertion.

The name need not be specified for a particular External_document.

If present, there shall be exactly one object that defines the name for an External_document.

4.2.163 External_document_assignment

An External_document_assignment is the assignment of an External_document (see 4.2.162) to an object that is described in this document.

The data associated with an External_document_assignment are the following:

- assigned_document;
- is_assigned_to;
- role.

4.2.163.1 assigned_document

The assigned_document specifies the External_document (see 4.2.162) that describes the object that is referenced by the External_document_assignment.

See 4.3.553 for the application assertion.

4.2.163.2 is_assigned_to

The is_assigned_to specifies the object that is described in the External_document (see 4.2.162).

See 4.3.545, 4.3.546, 4.3.547, 4.3.548, 4.3.549, 4.3.550, 4.3.551, 4.3.552, 4.3.554, 4.3.555, 4.3.556, 4.3.557, 4.3.558, 4.3.559, 4.3.560, 4.3.561, 4.3.562, 4.3.563, 4.3.564, 4.3.565, 4.3.566, 4.3.567, 4.3.568, 4.3.569, 4.3.570, 4.3.571, 4.3.572, 4.3.573, 4.3.574, 4.3.575, 4.3.576, and 4.3.577 for the application assertions.

There shall be exactly one object that the External_document_assignment is_assigned_to.

4.2.163.3 role

The role specifies the meaning of the External_document_assignment. Where applicable the following values shall be used:

- 'obligatory': The assigned product data shall conform to the content of the assigned External_document (see 4.2.162);
- 'informative': The assigned External_document may or may not be considered.

4.2.164 External_document_version

An External_document_version is a particular version of an External_document (see 4.2.162).

An External_document_version is a type of External_document (see 4.2.162).

The data associated with an External_document_version are the following:

- version_id.

The version_id specifies the identification of a particular version of an External_document (see 4.2.162).

4.2.165 External_library_reference

An External_library_reference is a mechanism to refer to an entry in an external library due to ISO 13584.

The data associated with an External_library_reference are the following:

- external_id.

The external_id specifies the unique identifier of the referenced entry in the external library. For an ISO 13584 compliant library this identification is a basic semantic unit according to ISO 13584–42.

4.2.166 External_model

An External_model is the identification of a model that is described in a Digital_document (see 4.2.139) and of the Cartesian_coordinate_space (see 4.2.44) that is needed to further process the externally described information.

The data associated with an External_model are the following:

- description;
- is_defined_as;
- is_defined_in;
- model_id;
- role.

4.2.166.1 description

The description specifies additional information about the External_model.

See 4.3.580 for the application assertion.

The description need not be specified for a particular External_model.

If present, there shall be exactly one object that defines the description for an External_model.

4.2.166.2 is_defined_as

The is_defined_as specifies the Digital_document (see 4.2.139) that contains the externally defined geometry information.

See 4.3.579 for the application assertion.

4.2.166.3 is_defined_in

The is_defined_in specifies the Cartesian_coordinate_space (see 4.2.44) that defines the context for the externally described geometry.

See 4.3.578 for the application assertion.

4.2.166.4 model_id

The model_id specifies the identifier of the External_model.

4.2.166.5 role

The role specifies the function performed by the External_model. Where applicable the following values shall be used:

- 'design shape': The geometry in the External_model represents the shape of an item as designed;

- 'idealized shape': The geometry in the External_model represents a simplified shape for e.g., analysis purposes.

4.2.167 Face

A Face is a topological object corresponding to a piece of a surface bounded by loops. A Face may have holes, each hole bounded by a loop. Associated with a Face is always a geometric surface. A Face has a normal N, and the tangent to a loop is T. For a loop bounding a Face, the cross product N $\times$ T points towards the interior of the Face.

A Face is a type of Detailed_geometric_model_element (see 4.2.136).

The data associated with a Face are the following:

- bound;
- face_geometry.

4.2.167.1 bound

The bound specifies the loop objects representing the boundaries of the Face.

See 4.3.581 for the application assertion.

4.2.167.2 face_geometry

The face_geometry specifies the basis geometry of the Face that is represented by a geometrical surface.

See 4.3.582 for the application assertion.

4.2.168 Face_bound

A Face_bound is a loop object defining the boundary of a Face (see 4.2.167).

The data associated with a Face_bound are the following:

- loop;
- outer.

4.2.168.1 loop

The loop specifies an Edge_loop (see 4.2.153) or a Vertex_loop (see 4.2.488) representing a boundary of a Face (see 4.2.167).

See 4.3.583 and 4.3.584 for the application assertions.

There shall be exactly one object that defines the loop for a Face_bound.

4.2.168.2 outer

The outer specifies wheter the loop defines the outer boundary of the Face (see 4.2.167) or not.

4.2.169 Face_termination

A Face_termination is a condition specifying that a Joggle_feature (see 4.2.250) extends from the rim of a panel to the next face the Joggle_feature intersects with.

A Face_termination is a type of Joggle_termination (see 4.2.251).

4.2.170 Face_transition

A Face_transition is the minimum state of geometric continuity between one Face (see 4.2.167) and its neighbour Face along one common edge.

The data associated with a Face_transition are the following:

- edge_joint;
- face_1;
- face_2;
- transition.

4.2.170.1 edge_joint

The edge_joint specifies the edge at the intersection of the two faces for the that Face_transition is describing the continuity.

NOTE – There may be cases where the two faces have more than one edge in common. In such a case the edge_joint is needed to identify for which of the edges the Face_transition is defined.

See 4.3.586 for the application assertion.

4.2.170.2 face_1

The face_1 specifies the first of the two neighbouring Face (see 4.2.167) objects.

See 4.3.585 for the application assertion.

4.2.170.3 face_2

The face_2 specifies the second of the two neighbouring Face (see 4.2.167) objects.

See 4.3.585 for the application assertion.

4.2.170.4 transition

The transition specifies the minimum state of geometric continuity from the first face to the second face along the intersecting edge by either a textual description or the definition of the required shape.

The following values shall be used:

- 'g0': G_0 continuity between two surfaces is a condition stating that the connected patches have common boundary curves [9];
- 'g1': G_1 continuity between two surfaces is a condition set on the two surface normals (in the tangency planes) along the boundary between the surfaces; for true G_1 continuity no angular difference is allowed [9];

– 'g2': G_2 continuity between two curves is a condition set on the radii of curvature of the surfaces along the boundary between the surfaces; for true G_2 continuity the radii along the boundary must be equal [9].

4.2.171 Faceted_b_rep

A Faceted_b_rep is defined by exactly one outer Faceted_closed_shell (see 4.2.173) and zero or more inner Faceted_closed_shell objects.

A Faceted_b_rep is a type of Detailed_geometric_model_element (see 4.2.136).

The data associated with a Faceted_b_rep are the following:

– outer;

– void.

4.2.171.1 outer

The outer specifies a Faceted_closed_shell (see 4.2.173) that defines the outer boundary of the Faceted_b_rep.

See 4.3.587 for the application assertion.

4.2.171.2 void

The void specifies a set of Faceted_closed_shell (see 4.2.173) objects that define internal cavities of a Faceted_b_rep.

NOTE – These shells have a normal pointing into the empty space inside.

See 4.3.587 for the application assertion.

4.2.172 Faceted_b_rep_model

A Faceted_b_rep_model is a collection of Faceted_b_rep (see 4.2.171) objects.

A Faceted_b_rep_model is a type of Geometric_model (see 4.2.211).

The data associated with a Faceted_b_rep_model are the following:

– element.

The element specifies a set of Faceted_b_rep (see 4.2.171) objects that form the Faceted_b_rep_model.

See 4.3.588 for the application assertion.

4.2.173 Faceted_closed_shell

A Faceted_closed_shell is a collection of at least four Faceted_face (see 4.2.174) that bounds a region in the three-dimensional space. The topological normal of the shell is defined as being

directed from the finite to the infinite region. A void is represented by an interior shell that builds a "hole" inside an outer shell. For such an interior shell the topological normal will point into the solid material. A Faceted_closed_shell shall not self-intersect.

The data associated with a Faceted_closed_shell are the following:

- shell_boundary.

The shell_boundary specifies the planar faces that represent the boundary of a closed shell.

See 4.3.589 for the application assertion.

4.2.174 Faceted_face

A Faceted_face is a topological object corresponding to a piece of a planar surface bounded by Poly_loop (see 4.2.330) objects. A Faceted_face may have holes, whereby each hole is bounded by a Poly_loop.

The data associated with a Faceted_face are the following:

- bound;
- face_geometry.

4.2.174.1 bound

The bound specifies the Poly_loop (see 4.2.330) objects representing the boundaries of the Faceted_face.

See 4.3.591 for the application assertion.

4.2.174.2 face_geometry

The face_geometry specifies the geometric representation of a planar surface that is represented by a Plane (see 4.2.317).

See 4.3.590 for the application assertion.

4.2.175 Feature_definition

A Feature_definition is a set of parameters and type information representing a feature that is either defined as a component of a part (Included_feature (see 4.2.232)) or independently in its own coordinate space for multiple use (Placed_feature (see 4.2.313)).

Each Feature_definition is either a Process_feature_in_solid (see 4.2.335), a Process_feature_in_panel (see 4.2.334), or a Replicate_feature (see 4.2.380).

The data associated with a Feature_definition are the following:

- is_defined_in;
- location_and_orientation.

4.2.175.1 is_defined_in

The is_defined_in specifies the Cartesian_coordinate_space_3d (see 4.2.46) the feature is defined in. In the case where the Feature_definition is used for defining an Included_feature (see 4.2.232) this Cartesian_coordinate_space_3d is identical with the coordinate space the corresponding part is defined in. In the case where the Feature_definition is used for defining a Placed_feature (see 4.2.313) this Cartesian_coordinate_space_3d is the feature's own local coordinate space.

See 4.3.593 for the application assertion.

4.2.175.2 location_and_orientation

The location_and_orientation specifies the Axis_placement (see 4.2.33) that corresponds to the location and orientation of the feature. The location_and_orientation shall only be used if the Feature_definition is used to define a feature that is applied as an Included_feature (see 4.2.232). In this case the Axis_placement is defined in the part's coordinate space.

NOTE – This kind of information is e.g. needed if the Feature_definition is used in a Replicate_feature (see 4.2.380).

See 4.3.592 for the application assertion.

The location_and_orientation need not be specified for a particular Feature_definition.

4.2.176 Feature_occurrence

A Feature_occurrence is the identification of a feature that is applied on a part.

NOTE – If a Feature_occurrence has shape information attached by a Shape_description_association (see 4.2.397) the representing model is defined in the same coordinate space as the Item_shape (see 4.2.245) that is referenced as 'composition'.

A Feature_occurrence is a type of Shape_aspect (see 4.2.394).

Each Feature_occurrence is either an Included_feature (see 4.2.232) or a Placed_feature (see 4.2.313).

The data associated with a Feature_occurrence are the following:

- feature_function.

The feature_function specifies the rationale of the application of a feature.

EXAMPLE 92 – The feature_function performed by a particular slot may be e.g., 'guide channel'.

See 4.3.594 for the application assertion.

The feature_function need not be specified for a particular Feature_occurrence.

If present, there shall be exactly one object that defines the feature_function for a Feature_occurrence.

4.2.177 Feature_parameter

A Feature_parameter is the representation of a characteristic of a Feature_definition (see 4.2.175).

The data associated with a Feature_parameter are the following:

- parameter_name;
- specified_value.

4.2.177.1 parameter_name

The parameter_name specifies the character, abbreviation, or word by which the Feature_-parameter is referred to.

EXAMPLE 93 – 'a', 'b1', and 'r' are typical examples for the parameter_name.

The parameter_name need not be specified for a particular Feature_parameter.

4.2.177.2 specified_value

The specified_value specifies the Value_with_unit (see 4.2.485) that has the actual value of the represented characteristic.

See 4.3.595 for the application assertion.

4.2.178 Feature_placement

A Feature_placement is the information about a geometric Transformation (see 4.2.472) and about a Geometric_model (see 4.2.211) that has the reference coordinate space used to define the placement of a Placed_feature (see 4.2.313).

The data associated with a Feature_placement are the following:

- placement_reference;
- transformation.

4.2.178.1 placement_reference

The placement_reference specifies the Geometric_model (see 4.2.211) that has the information about the reference coordinate space.

NOTE – A Placed_feature (see 4.2.313) may be placed relative to an Item (see 4.2.236) that is represented by a Geometric_model or it may be placed relative to another Placed_feature.

See 4.3.596 and 4.3.597 for the application assertions.

There shall be exactly one object that defines the placement_reference for a Feature_placement.

4.2.178.2 transformation

The transformation specifies the geometric Transformation (see 4.2.472) that is used to place a Placed_feature (see 4.2.313) in a reference coordinate space.

See 4.3.598 for the application assertion.

4.2.179 Featured_shape

A Featured_shape is the shape of an Item (see 4.2.236) that includes features. There shall be at least one Feature_occurrence (see 4.2.176) that is associated with the Featured_shape.

NOTE – The Featured_shape may be used to create a sequence of Item_shape (see 4.2.245) objects where each single Item_shape defines the shape of the Item at a different stage of manufacturing.

EXAMPLE 94 – A sequence of Item_shape respectively Featured_shape objects may consist of the following: Block without feature – Block with pocket – Block with pocket and hole.

A Featured_shape is a type of Item_shape (see 4.2.245).

The data associated with a Featured_shape are the following:

- base_shape.

The base_shape specifies the Item_shape (see 4.2.245) that defines the shape of the Item (see 4.2.236) before the application of the Feature_occurrence (see 4.2.176) objects that are associated with the Featured_shape.

See 4.3.599 for the application assertion.

4.2.180 Fill_area

A Fill_area is a bounded area containing colouring, hatching, or tiling that indicate its extent and content. A Fill_area communicates some aspect of a physical part characteristic or distinguishes it from its surroundings.

EXAMPLE 95 – Figure 26 depicts some fill area styles.

A Fill_area is a type of Annotation_element (see 4.2.15).

Each Fill_area may be an Associated_fill_area (see 4.2.30).

The data associated with a Fill_area are the following:

- boundary;
- reference_point.

4.2.180.1 boundary

The boundary specifies the curve bounding a particular fill area. The curve shall be closed and shall not self-intersect and it shall represent a planar 2D or 3D curve of the geometric model.

See 4.3.600 for the application assertion.

4.2.180.2 reference_point

The reference_point specifies a point within the fill area used in the placement and initiation of the fill area appearance. The reference point is a point through that a line of a hatching pattern shall pass or at which the origin of a tile shall be located. The reference point also establishes

the point at which the first visible segment of a line font used as the curve appearance for a hatching pattern starts.

The reference_point need not be specified for a particular Fill_area.

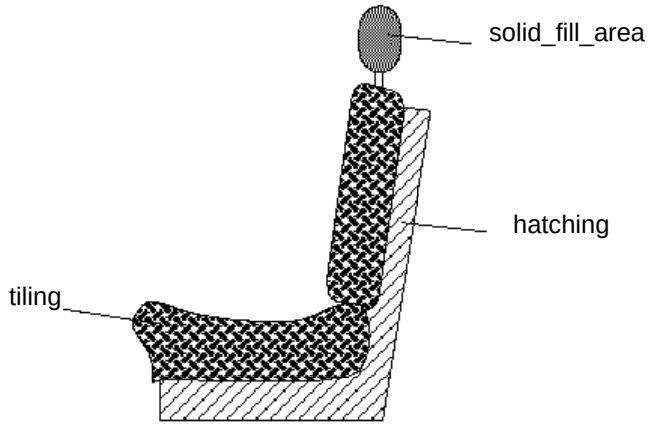


Figure 26 – Fill area styles

4.2.181 Fillet

A Fillet is a concave circular arc transition between two intersecting Face (see 4.2.167) objects without any constraints concerning changes of the radius along the Fillet.

A Fillet is a type of Transition_feature (see 4.2.476).

Each Fillet may be a Constant_radius_fillet (see 4.2.93).

4.2.182 Final_item_solution

A Final_item_solution is the specification of a set of Item_instance (see 4.2.241) objects that fulfill the same functional requirements as the neutral base element and have additional sensual characteristics.

EXAMPLE 96 – In the case of a car body the neutral parts are parts without paint; the final parts are the parts with paint. In the case of a seat the neutral part is the seat with unspecified cover material; the final part may be the seat covered with red leather.

A Final_item_solution is a type of Item_solution (see 4.2.246).

The data associated with a Final_item_solution are the following:

- final_item_status;
- final_specification.

4.2.182.1 final_item_status

The final_item_status specifies the level of completion between the neutral part and the final part.

EXAMPLE 97 – 'final for shipping overseas', 'final for transport by truck', or 'final for sale' are examples for the final_item_status.

4.2.182.2 final_specification

The final_specification specifies the means of finalization that is applied to the neutral part and which may either be a Descriptive_specification (see 4.2.129) or a Design_discipline_item_definition (see 4.2.134).

EXAMPLE 98 – The final_specification may be e.g., the Design_discipline_item_definition of a certain kind of paint.

NOTE – The Design_discipline_item_definition that is referenced here is not a view of the item that results from the finalization. The resulting, e.g. coloured, items may be identified by the Item_instance (see 4.2.241) objects that are associated with the Final_item_solution by the Solution_instance_association (see 4.2.413).

See 4.3.601 and 4.3.602 for the application assertions.

There shall be at least one object that defines the final_specification for a Final_item_solution.

4.2.183 Flat_bottom

A Flat_bottom is a hole bottom that is flat.

A Flat_bottom is a type of Hole_bottom_condition (see 4.2.228).

4.2.184 Flat_slot_end_type

A Flat_slot_end_type is an end condition of a slot that is a planar face normal to the sweep path at its start or end point respectively. The intersection of the slot wall surfaces and the end planar shape may but need not be blended by a radius.

EXAMPLE 99 – Figure 27 depicts a Flat_slot_end_type.

A Flat_slot_end_type is a type of Slot_end_type (see 4.2.406).

The data associated with a Flat_slot_end_type are the following:

- first_radius;
- second_radius.

4.2.184.1 first_radius

The first_radius specifies the constant amount of curvature of the first of two corners of the Flat_slot_end_type. The circle that defines the first radius is orientated normal to the first corner at

each point of the corner. The first corner of a Flat_slot_end_type is located where the slot wall intersects the slot end at a positive value on the x-axis of the defining Open_profile (see 4.2.292). The sum of the values of the two radii shall not exceed the smallest width of the Slot_feature (see 4.2.407).

NOTE – For this definition the profile and its x-axis is considered as 'moving' along the sweep path from one end of the slot to the other.

See 4.3.603 for the application assertion.

4.2.184.2 second_radius

The second_radius specifies the constant amount of curvature of the second of two corners of the Flat_slot_end_type. The circle that defines the first radius is orientated normal to the first corner at each point of the corner. The second corner of a Flat_slot_end_type is located where the slot wall intersects the slot end at a negative value on the x-axis of the defining Open_profile (see 4.2.292). The sum of the values of the two radii shall not exceed the smallest width of the Slot_feature (see 4.2.407).

NOTE – For this definition the profile and its x-axis is considered as 'moving' along the sweep path from one end of the slot to the other.

See 4.3.603 for the application assertion.

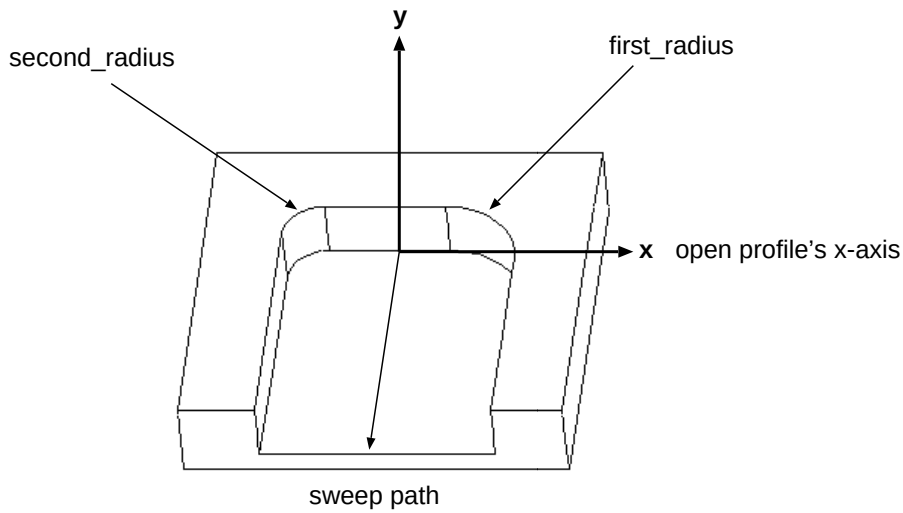


Figure 27 – Flat_slot_end_type

4.2.185 Flat_with_chamfer_bottom

A Flat_with_chamfer_bottom is a hole bottom that is flat and has a chamfer along the floor edge.

A Flat_with_chamfer_bottom is a type of Hole_bottom_condition (see 4.2.228).

The data associated with a Flat_with_chamfer_bottom are the following:

– chamfer.

The chamfer specifies the Chamfer_definition (see 4.2.52) that defines the chamfer of the hole bottom.

See 4.3.604 for the application assertion.

4.2.186 Flat_with_radius_bottom

A Flat_with_radius_bottom is a hole bottom that is flat and has a corner radius that is smaller than the radius of the concerned Round_hole (see 4.2.387).

A Flat_with_radius_bottom is a type of Hole_bottom_condition (see 4.2.228).

The data associated with a Flat_with_radius_bottom are the following:

– radius.

The radius specifies the corner radius between the side and the floor of the Round_hole (see 4.2.387).

See 4.3.605 for the application assertion.

4.2.187 Flatness_tolerance

A Flatness_tolerance is the limitation of the amount of deviation the elements of a surface may have from lying in one plane. It defines a tolerance zone that is two parallel planes within which the surface must lie. The Flatness_tolerance applies to the complete surface or to each segment of a defined size if the segment attribute is specified.

A Flatness_tolerance is a type of Geometric_tolerance (see 4.2.216).

The data associated with a Flatness_tolerance are the following:

– segment_size.

The segment_size specifies the length (or the edge length) of a segment if the geometric tolerance is not applied to the total length/area of an axis/surface but on a unit base (e.g. to each segment with a length of 20 mm).

The segment_size need not be specified for a particular Flatness_tolerance.

4.2.188 Form_feature_in_panel

A Form_feature_in_panel is the definition of a local modification in the shape of a panel that consists of a base boundary on the panel surface and of a target face that is a result of stamping the area that is enclosed by the base boundary. The Form_feature_in_panel is specified either by two Closed_profile_usage (see 4.2.72) objects or by a Closed_profile_usage and an Angle_specification (see 4.2.9). The use of two Angle_specification objects is not valid. In the case of a Form_feature_in_panel that has its base boundary defined by a Closed_profile (see 4.2.71) the placement of the feature shall be in a way that the base profile lies on the surface of the original panel.

The wall of the feature is a ruled surface for which the following rule applies: If P_b is a point on the base, P_t the corresponding point on the target, T_b the tangent to the base profile in P_b, and V the surface normal in P_b or the punch direction respectively, then the scalar product $(P_t - P_b) \cdot (V \times (T_b \times V))$ shall evaluate to zero.

A Form_feature_in_panel is a type of Process_feature_in_panel (see 4.2.334).

The data associated with a Form_feature_in_panel are the following:

- base_boundary_defined_by;
- base_fillet_radius;
- feature_type;
- target_boundary_defined_by;
- target_face;
- target_fillet_radius.

4.2.188.1 base_boundary_defined_by

The base_boundary_defined_by specifies either a Closed_profile_usage (see 4.2.72) that is a Closed_profile (see 4.2.71) on the panel surface or an Angle_specification (see 4.2.9) that defines the base boundary by applying an angle to the target boundary.

See 4.3.606 and 4.3.607 for the application assertions.

There shall be exactly one object that defines the base_boundary_defined_by for a Form_feature_in_panel.

4.2.188.2 base_fillet_radius

The base_fillet_radius specifies the blend of the edge where the Form_feature_in_panel meets the panel surface. The sum of the values of the base fillet radius and target fillet radius shall not exceed the minimum distance between the profile defining the base of the feature and the profile defining the target of the feature.

See 4.3.608 for the application assertion.

4.2.188.3 feature_type

The feature_type specifies the kind of Form_feature_in_panel that results from the application of the specified parameters. The feature_type shall be either 'pocket' or 'boss' or 'stairstep'.

NOTE – Usually a pocket in panel and a boss in panel don't differ in their shape but only in the function they perform.

4.2.188.4 target_boundary_defined_by

The target_boundary_defined_by specifies either a Closed_profile_usage (see 4.2.72) that corresponds to the boundary of the target face or an Angle_specification (see 4.2.9) that defines the target boundary by applying an angle to the base boundary.

See 4.3.606 and 4.3.607 for the application assertions.

There shall be exactly one object that defines the target_boundary_defined_by for a Form_feature_in_panel.

4.2.188.5 target_face

The target_face specifies the Surface (see 4.2.434) or the Shell (see 4.2.398) that defines that end of the Form_feature_in_panel that results from its application. The geometric elements representing the target shall be defined in the same Cartesian_coordinate_space (see 4.2.44) as the Form_feature_in_panel. In the case of a Form_feature_in_panel defined by a base profile and an Angle_specification (see 4.2.9) the target_face shall also be specified.

See 4.3.609 and 4.3.610 for the application assertions.

The target_face need not be specified for a particular Form_feature_in_panel.

If present, there shall be exactly one object that defines the target_face for a Form_feature_in_panel.

4.2.188.6 target_fillet_radius

The target_fillet_radius specifies the blend of the edge where the feature walls meet the target face. The sum of the values of the base fillet radius and target fillet radius shall not exceed the minimum distance between the profile defining the base of the feature and the profile defining the target of the feature.

See 4.3.608 for the application assertion.

4.2.189 Function_component_association

A Function_component_association is a mechanism to associate a Product_function (see 4.2.352) with a Product_component (see 4.2.350).

The data associated with a Function_component_association are the following:

- product_component;
- product_function.

4.2.189.1 product_component

The product_component specifies the Product_component (see 4.2.350) that fulfills a function defined by the Product_function (see 4.2.352).

See 4.3.611 for the application assertion.

4.2.189.2 product_function

The product_function specifies the Product_function (see 4.2.352) that is fulfilled by the Product_component (see 4.2.350).

See 4.3.612 for the application assertion.

4.2.190 General_boss

A General_boss is a volume adding feature that is specified by an enclosed volume bounded by an arbitrary shape.

EXAMPLE 100 – Figure 28 shows an example of a General_boss.

A General_boss is a type of Boss_feature (see 4.2.42).

The data associated with a General_boss are the following:

– boundary.

The boundary specifies the General_closed_profile (see 4.2.194) that is swept along a Linear_path (see 4.2.268) in order to create the General_boss.

See 4.3.613 for the application assertion.

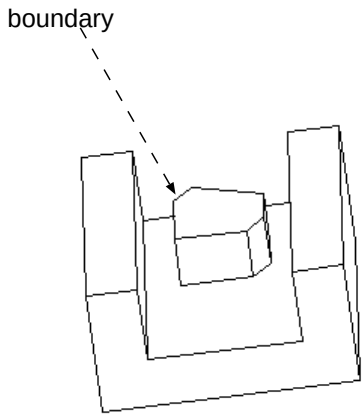


Figure 28 – General_boss

4.2.191 General_bottom_condition

A General_bottom_condition is a bottom of a Pocket_feature_in_solid (see 4.2.321) that is an enclosed area bounded by arbitrary non-planar shape.

A General_bottom_condition is a type of Pocket_bottom_condition (see 4.2.320).

The data associated with a General_bottom_condition are the following:

- floor_face;
- floor_radius.

4.2.191.1 floor_face

The floor_face specifies the Surface (see 4.2.434) or the Shell (see 4.2.398) that defines the bottom of the pocket. The geometric elements representing the floor shall be defined in the same Cartesian_coordinate_space (see 4.2.44) as the concerned Pocket_feature_in_solid (see 4.2.321).

See 4.3.615 and 4.3.616 for the application assertions.

There shall be exactly one object that defines the floor_face for a General_bottom_condition.

4.2.191.2 floor_radius

The floor_radius specifies the blend radius at the bottom edges of the Pocket_feature_in_solid (see 4.2.321).

See 4.3.614 for the application assertion.

The floor_radius need not be specified for a particular General_bottom_condition.

4.2.192 General_classification

A General_classification is a classification of an Item (see 4.2.236) which characterizes all items of the same kind independent from their usage.

EXAMPLE 101 – A fastener with subclasses, such as fastener with 2 or 3 fixture holes, a bracket with subclasses, e.g., with 90 or 100 degrees bending angle, or screws with subclasses such as metal screws or machine screws are examples for General_classification.

EXAMPLE 102 – For facilitating an effective part retrieval for slotted screws several General_classification objects to classify screws may be defined:

– A General_classification *1 with name 'slotted screws' and three subclasses with the names *2: 'slotted cheese head screws - product grade A', *3: 'slotted pan head screws - product grade A', and *4: 'slotted countersunk flat head screws - product grade A'.

– Each of the General_classification objects is based on a Classification_system (see 4.2.70) that further defines where the definition of the General_classification comes from. For General_classification *1 this is a company internal Classification_system for 'screw classes'. For the other General_classification objects these are ISO standardized Classification_system objects, namely for *2 'ISO 1207', for *3 'ISO 1580', and for *4 'ISO 2009'.

– The General_classification objects may be defined as well in an ISO 13584 compliant library (PLIB). In this case General_classification *1 references as a classification source an External_library_reference (see 4.2.165) that has an id corresponding to the class_BSU, i. e. a world wide unique identification of that class as defined according to ISO 13584–42. In another case the source of a General_classification may only be available as a document in paper form. In this case General_classification *4 references an Archived_document (see 4.2.24) with document id 'DIN EN ISO 2009' and name 'slotted countersunk flat head screws - product grade A'.

– The General_classification objects may be characterized further by Classification_attribute (see 4.2.69) objects. According to ISO 1207 the General_classification *2 has the attributes *11 with id 'd' and name 'thread', *12 with id 'l' and name 'nominal length', *13 with id 'mechanical properties' and name 'property class', etc. The possible values 'M1.6', 'M2', ..., 'M10' for Classification_attribute *11 may be specified through the set attribute values.

– To identify the designation 'Cheese head screw ISO 1207 - M5 x 20 x 4.8' a subclass of General_classification *2 is defined with the same name and classification source and the name 'slotted cheese head screws ISO 1207 -M5 x 20 x 4.8' but with the values for Classification_attribute corresponding to *11 restricted to 'M5' only, i.e. to get just the M5 screws, corresponding to *12 restricted to value '20' only, and corresponding to *13 restricted to value '4.8' only (for steel according to ISO 898–1).

A General_classification is a type of Item_classification (see 4.2.237).

The data associated with a General_classification are the following:

- attribute;
- classification_source;
- name;
- used_classification_system.

4.2.192.1 attribute

The attribute specifies the Classification_attribute (see 4.2.69) that represents one characteristic of the General_classification.

See 4.3.617 for the application assertion.

4.2.192.2 classification_source

The classification_source specifies the External_document (see 4.2.162) or the External_library_reference (see 4.2.165) that contains the specification of the General_classification. The External_library_reference (see 4.2.165) specifies the unique identification of a class, i. e. a class_BSU according to ISO 13584–42, for an ISO 13584 compliant library.

See 4.3.619 and 4.3.620 for the application assertions.

The classification_source need not be specified for a particular General_classification.

If present, there shall be exactly one object that defines the classification_source for a General_classification.

4.2.192.3 name

The name specifies the word or group of words by which the General_classification is referred to.

EXAMPLE 103 – 'fastener with 2 fixture holes' or 'bracket with 90 degrees bending angle' are examples for at.

See 4.3.621 for the application assertion.

There shall be exactly one object that defines the name for a General_classification.

4.2.192.4 used_classification_system

The used_classification_system specifies the Classification_system (see 4.2.70) that contains the information about how to interpret the name of the General_classification.

See 4.3.618 for the application assertion.

The used_classification_system need not be specified for a particular General_classification.

4.2.193 General_classification_hierarchy

A General_classification_hierarchy is used to build up hierarchical structures of General_classification (see 4.2.192) objects.

EXAMPLE 104 – With General_classification (see 4.2.192) part families such as screws can be classified. Additionally subclasses can be built and linked to the super-category screws, such as machine screw or self-tapping screw.

The data associated with a General_classification_hierarchy are the following:

- sub_classification;
- super_classification.

4.2.193.1 sub_classification

The sub_classification specifies the lower General_classification (see 4.2.192) in a General_classification_hierarchy that is included in the super class.

See 4.3.622 for the application assertion.

4.2.193.2 super_classification

The super_classification specifies the higher General_classification (see 4.2.192) in a General_classification_hierarchy that includes the sub class.

See 4.3.622 for the application assertion.

4.2.194 General_closed_profile

A General_closed_profile is a profile that is specified by an arbitrarily shaped enclosed area bounded by a of Curve (see 4.2.102). A General_closed_profile shall not have an additional Axis_placement (see 4.2.33) serving as parameter_reference. The set of Curve objects shall be

defined in the Cartesian_coordinate_space (see 4.2.44) that is referenced by the General_closed_-profile.

EXAMPLE 105 – Figure 29 illustrates a General_closed_profile.

A General_closed_profile is a type of Closed_profile (see 4.2.71).

The data associated with a General_closed_profile are the following:

– profile_shape.

The profile_shape specifies the Curve (see 4.2.102) that defines the closed profile shape.

See 4.3.623 for the application assertion.

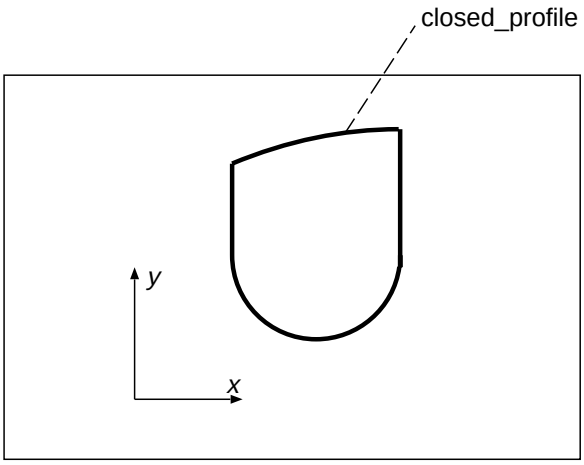


Figure 29 – General_closed_profile

4.2.195 General_compound_feature

A General_compound_feature is a General_feature (see 4.2.196) that is composed of several other features.

A General_compound_feature is a type of General_feature (see 4.2.196).

The data associated with a General_compound_feature are the following:

– elements.

The elements specifies the features that form the General_compound_feature.

See 4.3.624, 4.3.625, 4.3.626, and 4.3.627 for the application assertions.

There shall be at least two objects that defines the elements for a General_compound_feature.

4.2.196 General_feature

A General_feature is a type of feature that provides a mechanism to categorize an area of interest in a shape and to associate additional information, not intended for shape generation, with the identified aspect of the shape.

Each General_feature may be a General_compound_feature (see 4.2.195).

The data associated with a General_feature are the following:

- feature_category;
- feature_parameter;
- feature_type;
- location_and_orientation.

4.2.196.1 feature_category

The feature_category specifies the class to which the General_feature belongs.

NOTE – The values of feature_category are company specific.

EXAMPLE 106 – 'design feature', 'machining feature', and 'assembly feature' are examples for feature_category.

4.2.196.2 feature_parameter

The feature_parameter specifies a General_feature_parameter (see 4.2.197) that gives additional information about particular aspects of the General_feature.

See 4.3.629 for the application assertion.

4.2.196.3 feature_type

The feature_type specifies the kind of feature that is represented by the General_feature.

EXAMPLE 107 – 'planar feature' and 'logo feature' are examples for feature_type.

4.2.196.4 location_and_orientation

The location_and_orientation specifies an Axis_placement (see 4.2.33) that represents the location and the orientation of the General_feature in the part's coordinate space. Since the occurrence of a General_feature is an Included_feature (see 4.2.232) this Axis_placement as well as the shape representation of the General_feature is defined in the part's coordinate space.

NOTE – This Axis_placement is not a local coordinate system of the General_feature that might be used for geometric transformations.

See 4.3.628 for the application assertion.

The location_and_orientation need not be specified for a particular General_feature.

4.2.197 General_feature_parameter

A General_feature_parameter is a parameter describing a particular aspect of a General_feature (see 4.2.196).

The data associated with a General_feature_parameter are the following:

- parameter_role;
- parameter_value.

4.2.197.1 parameter_role

The parameter_role specifies the kind of parameter that is represented by the General_feature_-parameter.

EXAMPLE 108 – 'width', 'diameter', or 'font size' are examples for the parameter_role.

See 4.3.631 for the application assertion.

There shall be exactly one object that defines the parameter_role for a General_feature_parameter.

4.2.197.2 parameter_value

The parameter_value specifies the Feature_parameter (see 4.2.177) that has the value, the unit, and possibly a name of the General_feature_parameter.

See 4.3.630 for the application assertion.

4.2.198 General_item_definition_instance_relationship

A General_item_definition_instance_relationship is a relationship between a Design_discipline_-item_definition (see 4.2.134) and an Item_instance (see 4.2.241) whose meaning is defined by the attribute 'relation_type'.

EXAMPLE 109 – The relationship between the definition of a rubber gasket and the instances of the doors the gasket is used for is an example for a General_item_definition_instance_relationship. In this case the relationship gives information that the length of the gasket depends on the actual shape of each door.

A General_item_definition_instance_relationship is a type of Item_definition_instance_relationship (see 4.2.239).

The data associated with a General_item_definition_instance_relationship are the following:

- description;
- name;
- relation_type.

4.2.198.1 description

The description specifies additional information about the General_item_definition_instance_-relationship.

See 4.3.632 for the application assertion.

The description need not be specified for a particular General_item_definition_instance_relationship.

If present, there shall be exactly one object that defines the description for a General_item_definition_instance_relationship.

4.2.198.2 name

The name specifies the word or group of words by which the General_item_definition_instance_relationship is referred to.

See 4.3.632 for the application assertion.

There shall be exactly one object that defines the name for a General_item_definition_instance_relationship.

4.2.198.3 relation_type

The relation_type specifies the meaning of the relationship.

4.2.199 General_item_definition_relationship

A General_item_definition_relationship is a relationship between two Design_discipline_item_definition (see 4.2.134) objects whose meaning is defined by the attribute 'relation_type'.

EXAMPLE 110 – The functional dependency between that part of a door lock mounted on the door and the other part mounted on the car body is an example for a General_item_definition_relationship.

A General_item_definition_relationship is a type of Item_definition_relationship (see 4.2.240).

The data associated with a General_item_definition_relationship are the following:

- description;
- name;
- relation_type.

4.2.199.1 description

The description specifies additional information about the General_item_definition_relationship.

See 4.3.633 for the application assertion.

The description need not be specified for a particular General_item_definition_relationship.

If present, there shall be exactly one object that defines the description for a General_item_definition_relationship.

4.2.199.2 name

The name specifies the word or group of words by which the General_item_definition_relationship is referred to.

See 4.3.633 for the application assertion.

There shall be exactly one object that defines the name for a General_item_definition_relationship.

4.2.199.3 relation_type

The relation_type specifies the meaning of the relationship.

4.2.200 General_item_instance_relationship

A General_item_instance_relationship is a relationship between two Item_instance (see 4.2.241) objects whose meaning is defined by the attribute 'relation_type'.

EXAMPLE 111 – The relationship between a door and the adjacent parts of the car body (like A-column or roof) that defines the actual gap is an example for a General_item_instance_relationship.

A General_item_instance_relationship is a type of Item_instance_relationship (see 4.2.242).

The data associated with a General_item_instance_relationship are the following:

- description;
- name;
- relation_type.

4.2.200.1 description

The description specifies additional information about the General_item_instance_relationship.

See 4.3.634 for the application assertion.

The description need not be specified for a particular General_item_instance_relationship.

If present, there shall be exactly one object that defines the description for a General_item_instance_relationship.

4.2.200.2 name

The name specifies the word or group of words by which the General_item_instance_relationship is referred to.

See 4.3.634 for the application assertion.

There shall be exactly one object that defines the name for a General_item_instance_relationship.

4.2.200.3 relation_type

The relation_type specifies the meaning of the relationship.

4.2.201 General_item_property

A General_item_property is the definition of an item's property that is specified by the attribute 'property type'.

NOTE – The General_item_property allows to attach geometric information to an item; yet, this geometry is not supposed to represent the shape of the item.

EXAMPLE 112 – The noise emission of an engine, maintenance intervals, wheel base, and duration of parts, or electrical properties are examples for a General_item_property.

A General_item_property is a type of Item_property (see 4.2.243).

The data associated with a General_item_property are the following:

- description;
- property_type;
- property_value.

4.2.201.1 description

The description specifies additional information about the General_item_property.

See 4.3.637 for the application assertion.

The description need not be specified for a particular General_item_property.

If present, there shall be exactly one object that defines the description for a General_item_-property.

4.2.201.2 property_type

The property_type specifies the kind of property the General_item_property defines.

4.2.201.3 property_value

The property_value specifies the value of the General_item_property. The value of a General_-item_property is either a descriptive value, a numeric value, or a piece of geometry.

See 4.3.635, 4.3.636, 4.3.638, and 4.3.639 for the application assertions.

There shall be at least one object that defines the property_value for a General_item_property.

4.2.202 General_open_profile

A General_open_profile is a profile that is specified by an arbitrarily shaped open boundary. The area of interest is on the convex side of the General_open_profile (in the case of a removal feature the convex side is removed; in the case of a protrusion feature the convex side is added).

A General_open_profile shall not have an additional Axis_placement (see 4.2.33) serving as parameter_reference. The defining Curve is always defined in the Cartesian_coordinate_space (see 4.2.44) that is referenced by the General_open_profile as 'is_defined_in'.

EXAMPLE 113 – Figure 30 illustrates a General_open_profile.

A General_open_profile is a type of Open_profile (see 4.2.292).

The data associated with a General_open_profile are the following:

– profile_shape.

The profile_shape specifies the Curve (see 4.2.102) that defines the open profile shape.

See 4.3.640 for the application assertion.

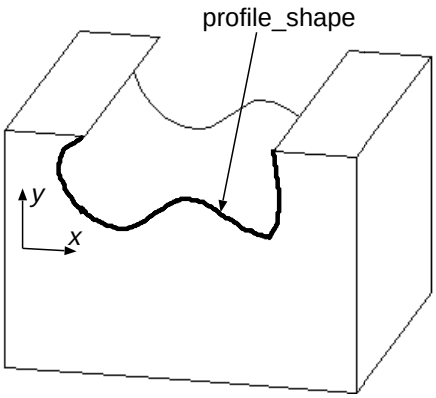


Figure 30 – General_open_profile

4.2.203 General_path

A General_path is a path that is specified by a direction of travel along an arbitrary shape.

EXAMPLE 114 – Figure 31 illustrates a General_path.

A General_path is a type of Path (see 4.2.304).

The data associated with a General_path are the following:

– path_geometry.

The path_geometry specifies the Curve (see 4.2.102) that defines the General_path. These Curve objects shall not intersect or duplicate themselves.

See 4.3.641 for the application assertion.

4.2.204 General_pattern

A General_pattern is a pattern that specifies a base shape component and a set of arbitrary placements to arrange identical copies of the base feature.

EXAMPLE 115 – Figure 32 depicts a General_pattern with a base shape and 5 added instances.

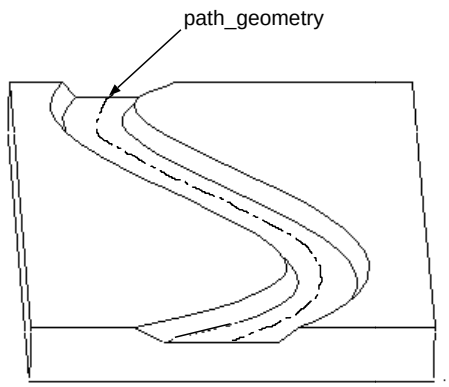


Figure 31 – General_path

A General_pattern is a type of Replicate_feature (see 4.2.380).

The data associated with a General_pattern are the following:

– added_instance.

The added_instance specifies the set of Axis_placement (see 4.2.33) objects that represent the locations and orientations of the additional instances of the base feature in the General_pattern.

The placement of the additional instances is performed by matching either the Axis_placement that serves as location for the Feature_definition (see 4.2.175) of the base feature (in the case of an Included_feature (see 4.2.232)) or its local coordinate system (in the case of a Placed_feature (see 4.2.313)) with the Axis_placement objects that identify the additional instances.

The Axis_placement objects are defined in the same Cartesian_coordinate_space (see 4.2.44) as the General_pattern.

NOTE – The base feature is the first instance in the pattern. In order to form a pattern one or more additional instances have to be specified.

See 4.3.642 for the application assertion.

4.2.205 General_pocket_in_solid

A General_pocket_in_solid is a pocket that is specified by a volume of arbitrary shape removed from the part.

EXAMPLE 116 – Figure 33 depicts examples of a General_pocket_in_solid.

A General_pocket_in_solid is a type of Pocket_feature_in_solid (see 4.2.321).

The data associated with a General_pocket_in_solid are the following:

– boundary.

The boundary specifies the enclosing bounds for the General_pocket_in_solid.

See 4.3.643 for the application assertion.

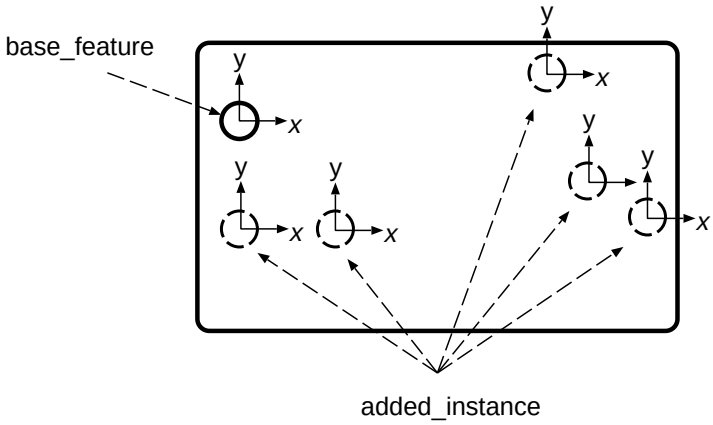


Figure 32 – General_pattern

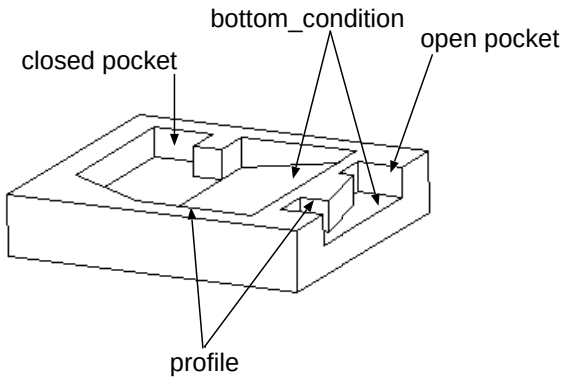


Figure 33 – General_pocket_in_solid

4.2.206 General_process_property

A General_process_property is the definition of a process's property that is specified by the attributes 'type'.

A General_process_property is a type of Process_property (see 4.2.345).

The data associated with a General_process_property are the following:

- description;
- property_type;
- property_value.

4.2.206.1 description

The description specifies additional information about the General_process_property.

See 4.3.646 for the application assertion.

The description need not be specified for a particular General_process_property.

If present, there shall be exactly one object that defines the description for a General_process_property.

4.2.206.2 property_type

The property_type specifies the kind of property the General_process_property defines.

4.2.206.3 property_value

The property_value specifies the value of the General_process_property. The value of a General_process_property is either a descriptive value, a numeric value, or a piece of geometry.

See 4.3.644, 4.3.645, 4.3.647, and 4.3.648 for the application assertions.

There shall be at least one object that defines the property_value for a General_process_property.

4.2.207 General_tolerance_table

A General_tolerance_table is a table containing information about General_tolerances (see 4.2.208).

The data associated with a General_tolerance_table are the following:

- tolerance_type.

The tolerance_type specifies the kind of the tolerance.

EXAMPLE 117 – Examples for tolerance_type are 'length dimensional tolerance', 'angular dimensional tolerance', or 'rectangularity'.

4.2.208 General_tolerances

A General_tolerances is a default value for length dimension and angular dimension tolerances. The default values are specified in a table form to select the correct tolerance type and values for a given dimension. These default values are overwritten by tolerances that are assigned explicitly to particular aspects of the shape. The concept of General_tolerances is defined in ISO 2768.

EXAMPLE 118 – The General_tolerances specifies, e.g., that for the tolerance class 'f' the tolerance value '±0.15' is applicable for length dimensions between 30 mm and 120 mm. Figure 34 depicts an example for General_tolerances.

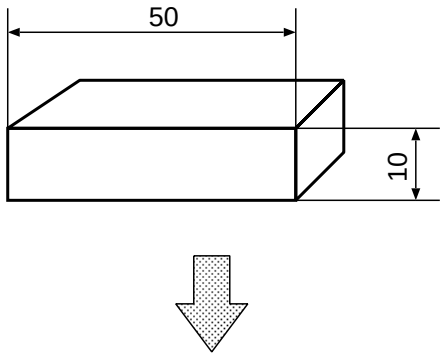
The data associated with a General_tolerances are the following:

- table_definition.

The table_definition specifies the contents of the General_tolerances that is defined either explicitly in a General_tolerance_table (see 4.2.207) or in an External_document (see 4.2.162).

See 4.3.649 and 4.3.650 for the application assertions.

There shall be exactly one object that defines the table_definition for a General_tolerances.



standard source: ISO 2768

length dimensional tolerance:

tolerance class	length nominal 0.5-3	3-6	6-30	30-120	120-400	etc.
f	0.05	0.05	0.1	0.15	0.2	
m	0.1	0.1	0.2	0.3	0.5	
c	0.2	0.3	0.5	0.8	1.2	
v	-	0.5	1.0	1.5	2.5	

tolerance ± mm

angular dimensional tolerance:

tolerance class	short angle length nominal 0-10	10-50	50-120	120-400	>400
f	1°	30'	20'	10'	5'
m	1°	30'	20'	10'	5'
c	1°30'	1°	30'	15'	10'
v	3°	2°	1°	30'	20'

tolerance ± o

rectangularity:

tolerance class	short angle length nominal 0-100	100-300	300-1000	1000-3000
H	0.2	0.3	0.4	0.5
K	0.4	0.6	0.8	1
L	0.6	1	1.5	2

tolerance ± mm

•
•
•

Figure 34 – General_tolerances

4.2.209 General_top_condition

A General_top_condition is a top condition of a boss that is bounded by arbitrary non-planar shape.

A General_top_condition is a type of Boss_top_condition (see 4.2.43).

The data associated with a General_top_condition are the following:

– top_face.

The top_face specifies the Surface (see 4.2.434) or the Shell (see 4.2.398) that defines the top of the boss. The geometric elements representing the top shall be defined in the same Cartesian_coordinate_space (see 4.2.44) as the concerned Boss_feature (see 4.2.42).

See 4.3.651 and 4.3.652 for the application assertions.

There shall be exactly one object that defines the top_face for a General_top_condition.

4.2.210 Geometric_dimension

A Geometric_dimension is the representation of the size of an element or of the distance between two elements.

NOTE – A Geometric_dimension may be presented by a Dimension_callout (see 4.2.141) showing the dimension value and its relationship to the drafted geometry or annotation.

Each Geometric_dimension is either a Size_dimension (see 4.2.405) or a Location_dimension (see 4.2.270).

The data associated with a Geometric_dimension are the following:

- dimension_value;
- id;
- name.

4.2.210.1 dimension_value

The dimension_value specifies the value of the size of the element or the distance between the two elements. The dimension_value shall not be specified if the Geometric_dimension is associated with a Value_limit (see 4.2.482) or with a Value_range (see 4.2.483). In all other cases the dimension_value shall be specified.

The dimension_value need not be specified for a particular Geometric_dimension.

4.2.210.2 id

The id specifies the identifier of the Geometric_dimension.

4.2.210.3 name

The name specifies the word or group of words by which the Geometric_dimension is referred to.

See 4.3.653 for the application assertion.

The name need not be specified for a particular Geometric_dimension.

If present, there shall be exactly one object that defines the name for a Geometric_dimension.

4.2.211 Geometric_model

A Geometric_model is a representation of geometry. A Geometric_model that does not reference any Detailed_geometric_model_element (see 4.2.136) objects through one the subtypes directly shall reference at least one Template_instance (see 4.2.453) as 'additional_elements'.

Each Geometric_model may be a Geometric_model_version (see 4.2.214).

Each Geometric_model is either a Wireframe_model_3d (see 4.2.498), a Point_direction_model (see 4.2.325), a Wireframe_model_2d (see 4.2.497), a Hybrid_geometric_model_3d (see 4.2.230), a Geometrically_bounded_surface_model (see 4.2.220), a Surface_model (see 4.2.439), a Compound_model (see 4.2.87), a Csg_model (see 4.2.100), a B_rep_model (see 4.2.35), or a Faceted_b_rep_model (see 4.2.172).

The data associated with a Geometric_model are the following:

- additional_elements;
- description;
- is_defined_in;
- model_extent;
- model_id;
- role.

4.2.211.1 additional_elements

The additional_elements specifies the Template_instance (see 4.2.453) objects that are additionally imported into the Geometric_model using a Transformation (see 4.2.472). The additional_elements shall also of the same type as the referring Geometric_model.

See 4.3.656 for the application assertion.

4.2.211.2 description

The description specifies additional information about the Geometric_model.

See 4.3.655 for the application assertion.

The description need not be specified for a particular Geometric_model.

If present, there shall be exactly one object that defines the description for a Geometric_model.

4.2.211.3 is_defined_in

The is_defined_in specifies the Cartesian_coordinate_space (see 4.2.44) in which the Geometric_model is defined. The specified Cartesian_coordinate_space serves also as the reference coordinate space for the transformation of Template_instance (see 4.2.453) objects used as additional elements in the Geometric_model.

See 4.3.654 for the application assertion.

4.2.211.4 model_extent

The model_extent specifies the radius of a sphere that contains all elements of the model and whose centre is at the origin of the Cartesian_coordinate_space (see 4.2.44) of the Geometric_model.

The model_extent need not be specified for a particular Geometric_model.

4.2.211.5 model_id

The model_id specifies the identifier of the Geometric_model.

4.2.211.6 role

The role specifies the function performed by the Geometric_model. Where applicable the following values shall be used:

- 'design shape': The geometry in the Geometric_model represents the shape of an item as designed;
- 'idealized shape': The geometry in the Geometric_model represents a simplified shape for e.g., analysis purposes.

4.2.212 Geometric_model_relationship

A Geometric_model_relationship is a relationship between two Geometric_model (see 4.2.211) objects.

Each Geometric_model_relationship may be a Geometric_model_relationship_with_transformation (see 4.2.213).

The data associated with a Geometric_model_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.212.1 description

The description specifies additional information about the Geometric_model_relationship.

See 4.3.659 for the application assertion.

The description need not be specified for a particular Geometric_model_relationship.

If present, there shall be exactly one object that defines the description for a Geometric_model_relationship.

4.2.212.2 related

The related specifies the second of the two Geometric_model (see 4.2.211) objects related by the Geometric_model_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.657 and 4.3.658 for the application assertions.

There shall be exactly one object that the Geometric_model_relationship is related to.

4.2.212.3 relating

The relating specifies the first of the two Geometric_model (see 4.2.211) objects related by the Geometric_model_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.657 and 4.3.658 for the application assertions.

There shall be exactly one object that the Geometric_model_relationship is relating to.

4.2.212.4 relation_type

The relation_type specifies the meaning of the relationship.

EXAMPLE 119 – The relation_type may be e.g., 'basis for' if the relating Geometric_model (see 4.2.211) represents the design shape of an object and the related Geometric_model represents an idealized shape of the same object.

4.2.213 Geometric_model_relationship_with_transformation

A Geometric_model_relationship_with_transformation is a relationship between two Geometric_-model (see 4.2.211) objects with the additional information about a geometric Transformation (see 4.2.472) that defines the location and orientation of the related model relative to the relating model.

A Geometric_model_relationship_with_transformation is a type of Geometric_model_relationship (see 4.2.212).

The data associated with a Geometric_model_relationship_with_transformation are the following:

- transformation.

The transformation specifies the geometric Transformation (see 4.2.472) that places and orients the related model relative to the relating model.

See 4.3.660 for the application assertion.

4.2.214 Geometric_model_version

A Geometric_model_version is a particular version of a Geometric_model (see 4.2.211).

A Geometric_model_version is a type of Geometric_model (see 4.2.211).

The data associated with a Geometric_model_version are the following:

- version_id.

The version_id specifies the identification of a particular version of a Geometric_model (see 4.2.211).

4.2.215 Geometric_set_3d

A Geometric_set_3d is a set of Curve (see 4.2.102), Point (see 4.2.322), or Surface (see 4.2.434) objects that are part of a Geometrically_bounded_surface_model (see 4.2.220).

A Geometric_set_3d is a type of Detailed_geometric_model_element (see 4.2.136).

The data associated with a Geometric_set_3d are the following:

- element.

The element specifies the set of geometric elements that are are included in the Geometric_set_3d.

See 4.3.661, 4.3.662, and 4.3.663 for the application assertions.

There shall be at least one object that defines the element for a Geometric_set_3d.

4.2.216 Geometric_tolerance

A Geometric_tolerance is the allowable deviation of physical characteristics according to ISO 1101.

Each Geometric_tolerance is either a Cylindricity_tolerance (see 4.2.109), a Symmetry_tolerance (see 4.2.444), a Straightness_tolerance (see 4.2.424), a Flatness_tolerance (see 4.2.187), a Position_tolerance (see 4.2.332), a Circular_runout_tolerance (see 4.2.62), a Linear_profile_tolerance (see 4.2.269), a Circularity_tolerance (see 4.2.63), a Surface_profile_tolerance (see 4.2.442), a Total_runout_tolerance (see 4.2.471), a Parallelism_tolerance (see 4.2.301), a Perpendicularity_tolerance (see 4.2.306), a Concentricity_tolerance (see 4.2.88), or an Angularity_tolerance (see 4.2.13).

The data associated with a Geometric_tolerance are the following:

- is_applied_to;
- is_controlled_by;
- name;
- qualifying_note;
- significant_digits;
- tolerance_value;
- value_interpretation.

4.2.216.1 is_applied_to

The is_applied_to specifies the shape that the Geometric_tolerance is associated to.

See 4.3.664, 4.3.665, 4.3.666, and 4.3.669 for the application assertions.

There shall be exactly one object that the Geometric_tolerance is_applied_to.

4.2.216.2 is_controlled_by

The is_controlled_by specifies the Material_condition (see 4.2.278) that is associated with the application of the Geometric_tolerance.

See 4.3.667 for the application assertion.

The is_controlled_by need not be specified for a particular Geometric_tolerance.

4.2.216.3 name

The name specifies the word or group of words by which the Geometric_tolerance is referred to.

See 4.3.668 for the application assertion.

The name need not be specified for a particular Geometric_tolerance.

If present, there shall be exactly one object that defines the name for a Geometric_tolerance.

4.2.216.4 qualifying_note

The qualifying_note specifies additional constraints on the Geometric_tolerance.

EXAMPLE 120 – 'not concave' is an example for the qualifying_note.

See 4.3.668 for the application assertion.

The qualifying_note need not be specified for a particular Geometric_tolerance.

If present, there shall be exactly one object that defines the qualifying_note for a Geometric_-tolerance.

4.2.216.5 significant_digits

The significant_digits specifies the number of decimal places indicating the required accuracy of the tolerance.

The significant_digits need not be specified for a particular Geometric_tolerance.

4.2.216.6 tolerance_value

The tolerance_value specifies the tolerance magnitude expressed by a measure with unit.

4.2.216.7 value_interpretation

The value_interpretation specifies information on how the Geometric_tolerance shall be interpreted.

NOTE – A Geometric_tolerance may be specified in the design stage of a product but it may also be documented as measured on a prototype.

EXAMPLE 121 – The value_interpretation may be e.g., 'required', 'designed', 'estimated', or 'measured'.

4.2.217 Geometric_tolerance_precedence

A Geometric_tolerance_precedence is a relationship between two Geometric_tolerance (see 4.2.216) objects indicating that one of the Geometric_tolerance objects is of higher priority than the other one.

EXAMPLE 122 – A part may have a shape that has a Geometric_tolerance and that is he basic shape for a pattern that may also have a Geometric_tolerance. In this case the tolerance for the base shape usually has precedence over the tolerance for the pattern.

The data associated with a Geometric_tolerance_precedence are the following:

- primary_tolerance;
- secondary_tolerance.

4.2.217.1 primary_tolerance

The primary_tolerance specifies the Geometric_tolerance (see 4.2.216) that has precedence.

See 4.3.670 for the application assertion.

4.2.217.2 secondary_tolerance

The secondary_tolerance specifies the Geometric_tolerance (see 4.2.216) that is of lower priority.

See 4.3.670 for the application assertion.

4.2.218 Geometric_tolerance_symbol

A Geometric_tolerance_symbol is a combination of annotation elements that, as collection, presents a Geometric_tolerance (see 4.2.216).

A Geometric_tolerance_symbol is a type of Associated_annotation_symbol (see 4.2.29).

4.2.219 Geometrical_relationship

A Geometrical_relationship is the relationship between two Design_discipline_item_definition (see 4.2.134) objects specifying two parts that are geometrically related. Therefore the shape of the

related parts has to be represented only once and can be reused for the geometrically related situation. The two Design_discipline_item_definition objects that are related by this relationship shall belong to different Item_version (see 4.2.248) objects. Other properties except shape are not subject to this relationship and therefore not 'inherited' from the relating Design_discipline_item_definition to the related Design_discipline_item_definition.

EXAMPLE 123 – Figure 35 shows an example for a Geometrical_relationship: In this case the left mudguard of a car is symmetrical to the right mudguard. A given plane mirroring transformation specifies the location of the left mudguard regarding the location of the right mudguard.

A Geometrical_relationship is a type of Item_definition_relationship (see 4.2.240).

The data associated with a Geometrical_relationship are the following:

- description;
- name;
- transformation.

4.2.219.1 description

The description specifies additional information about the Geometrical_relationship.

See 4.3.672 for the application assertion.

The description need not be specified for a particular Geometrical_relationship.

If present, there shall be exactly one object that defines the description for a Geometrical_relationship.

4.2.219.2 name

The name specifies the word or group of words by which the Geometrical_relationship is referred to.

See 4.3.672 for the application assertion.

The name need not be specified for a particular Geometrical_relationship.

If present, there shall be exactly one object that defines the name for a Geometrical_relationship.

4.2.219.3 transformation

The transformation specifies the Geometric_model_relationship_with_transformation (see 4.2.213) or the Template_instance (see 4.2.453) that has the Transformation (see 4.2.472) to be applied to the relating Design_discipline_item_definition (see 4.2.134) in order to define the location and orientation of the related Design_discipline_item_definition. Translation, rotation, and mirroring, i.e., inversion, is included; scaling is not included. In the case of a Template_instance the scale factor shall be omitted or set to 1.0.

See 4.3.671 and 4.3.673 for the application assertions.

There shall be exactly one object that defines the transformation for a Geometrical_relationship.

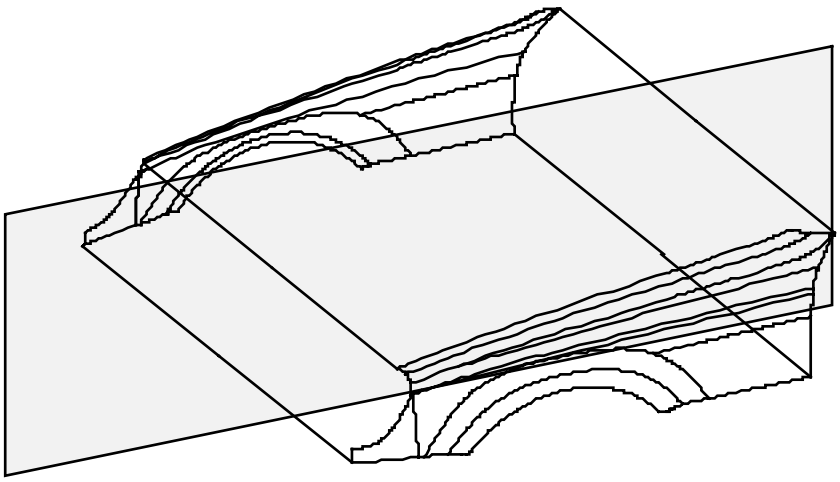


Figure 35 – Geometrical_relationship

4.2.220 Geometrically_bounded_surface_model

A Geometrically_bounded_surface_model is a surface model without topology. This model is formed by the use of points, curves and surfaces only. All unbounded curves and surfaces shall be explicitly trimmed.

A Geometrically_bounded_surface_model is a type of Geometric_model (see 4.2.211).

The data associated with a Geometrically_bounded_surface_model are the following:

- element.

The element specifies the set of Geometric_set_3d (see 4.2.215) objects that form the Geometrically_bounded_surface_model.

See 4.3.674 for the application assertion.

4.2.221 Ground_representation

A Ground_representation is the geometric representation of that part of the kinematic model that shall never move.

The data associated with a Ground_representation are the following:

- shape_description.

The shape_description specifies the shape model of the Ground_representation.

See 4.3.675 for the application assertion.

4.2.222 Group

A Group is a general structure for objects that can be nested, i.e., groups can contain other groups. The nesting shall be acyclic. In case of nesting, a Group need not contain elements.

EXAMPLE 124 – In a B-rep model several drilling holes are included. The objects defining the drilling holes including the two intersection curves and the faces of the hole may be grouped together with the purpose 'drilling hole'.

The data associated with a Group are the following:

- element;
- name;
- purpose.

4.2.222.1 element

The element specifies the Detailed_element (see 4.2.135) objects included in the Group.

See 4.3.676 for the application assertion.

4.2.222.2 name

The name specifies the word or group of words by which the Group is referred to.

See 4.3.677 for the application assertion.

There shall be exactly one object that defines the name for a Group.

4.2.222.3 purpose

The purpose specifies the rationale behind the grouping mechanism.

See 4.3.677 for the application assertion.

The purpose need not be specified for a particular Group.

If present, there shall be exactly one object that defines the purpose for a Group.

4.2.223 Group_relationship

A Group_relationship is a relationship between two Group (see 4.2.222) objects.

NOTE – The existence of a Group_relationship does not mean that the inclusion of an element in the related Group implies the inclusion of that element also in the relating Group.

The data associated with a Group_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.223.1 description

The description specifies additional information about the Group_relationship.

See 4.3.679 for the application assertion.

The description need not be specified for a particular Group_relationship.

If present, there shall be exactly one object that defines the description for a Group_relationship.

4.2.223.2 related

The related specifies the second of the two Group (see 4.2.222) objects that are related by the Group_relationship.

See 4.3.678 for the application assertion.

4.2.223.3 relating

The relating specifies the first of the two Group (see 4.2.222) objects that are related by the Group_relationship.

See 4.3.678 for the application assertion.

4.2.223.4 relation_type

The relation_type specifies the meaning of the relationship.

4.2.224 Half_space_solid

A Half_space_solid is the half space that is the regular subset of the domain that lies on one side of an unbounded surface. A Boolean flag shall be set to indicate whether the Half_space_solid is selected according to the normal direction of the surface. A Half_space_solid may participate in Boolean operations of the type 'intersection' or 'difference' as the second Boolean operand.

NOTE – Boxed half space solids are out of scope of this part of ISO 10303.

A Half_space_solid is a type of Detailed_geometric_model_element (see 4.2.136).

4.2.225 Hardness

A Hardness describes the hardness of a material on its surface. The hardness represents the resistance of a surface to deformation by external forces.

EXAMPLE 125 – Figure 36 gives examples for the specification of surface hardness on technical drawings. On the left side a surface hardness with the measuring method 'Vickers HV30' according to ISO 4516 [4], the hardness value '620', the hardness value tolerance '+160', the treatment type 'Rht' (surface layer hardening), the hardness limit '500', the hardening depth '0,8 mm', and the hardening depth tolerance '+0,8 mm' is specified. On the right side a surface hardness with the measuring method 'Rockwell HRC' according to ISO 6508 [5], the hardness value '57', the hardness

value tolerance '+6', the treatment type 'Eht' (case hardening), the hardening depth '1,2 mm', and the hardening depth tolerance '+0,7 mm' is specified.

A Hardness is a type of Surface_condition (see 4.2.437).

The data associated with a Hardness are the following:

- depth;
- description;
- environment_condition;
- hardness_value;
- measuring_method;
- measuring_position.

4.2.225.1 depth

The depth specifies the material depth measured from the part's surface into the material where the specified hardness applies.

EXAMPLE 126 – The depth of the hardened material after a high-frequency quenching process is an example for depth.

See 4.3.681 for the application assertion.

The depth need not be specified for a particular Hardness.

4.2.225.2 description

The description specifies additional information about the Hardness.

EXAMPLE 127 – 'depth is describing the total effective case depth' is an example for description.

See 4.3.680 for the application assertion.

The description need not be specified for a particular Hardness.

If present, there shall be exactly one object that defines the description for a Hardness.

4.2.225.3 environment_condition

The environment_condition specifies the conditions under that the specified hardness_value is valid.

There may be more than one environment_condition for a Hardness.

4.2.225.4 hardness_value

The hardness_value specifies the value with unit for the hardness where the actual values have to measured with the specified measuring method.

See 4.3.681 for the application assertion.

4.2.225.5 measuring_method

The measuring_method specifies the method for measuring hardness values.

NOTE – Standardized hardness measuring methods are Brinell, Rockwell, or Vickers.

4.2.225.6 measuring_position

The measuring_position specifies the position where the hardness is measured with respect to the treated part.

EXAMPLE 128 – 'core' or 'surface' are examples for measuring_position where 'core' is inside the hardened part, i.e., it is measured before a quenching process, and 'surface' is on the surface of a hardened part, i.e., it is measured after a quenching process.

The measuring_position need not be specified for a particular Hardness.

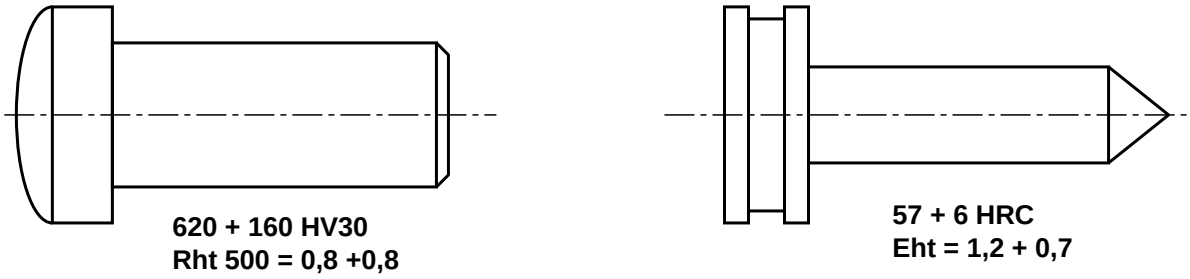


Figure 36 – Hardness

4.2.226 Hatching

A Hatching is a list of hatch patterns that are presented by single, uniformly spaced geometric pattern of lines. The basis of the hatching pattern is an infinite straight line that is repeated across the fill area and clipped to its boundaries. The Hatching may be defined by the user or may be externally defined.

4.2.227 Hidden_element_handling

A Hidden_element_handling is the definition how hidden elements are handled in a Model_image (see 4.2.284). The element types which may hide other elements are solid and surface type

geometric elements. The element types which may be hidden by these solids and surfaces are all geometric and annotation type elements.

The data associated with a Hidden_element_handling are the following:

- hidden_element_appearance;
- is_defined_for;
- switch.

4.2.227.1 hidden_element_appearance

The hidden_element_appearance specifies the appearance characteristics of hidden elements.

EXAMPLE 129 – A curve appearance with a dashed line type shows lines that are hidden in a dashed style instead of not showing them at all.

See 4.3.682 for the application assertion.

The hidden_element_appearance need not be specified for a particular Hidden_element_handling.

4.2.227.2 is_defined_for

The is_defined_for specifies the Model_image (see 4.2.284) that the Hidden_element_handling is applied to.

See 4.3.683 for the application assertion.

4.2.227.3 switch

The switch specifies the status of hiding for the hidden element handling. Valid values for switch are 'active' and 'inactive'. In the status 'inactive' all elements are displayed as usual. In the status 'active' the hidden elements are either not shown or visualized with a specialized appearance defined by the attribute hidden_element_appearance.

4.2.228 Hole_bottom_condition

A Hole_bottom_condition is the definition of a kind of end that characterizes a Round_hole (see 4.2.387).

EXAMPLE 130 – Figure 37 illustrates the various hole bottom conditions.

Each Hole_bottom_condition is either a Conical_bottom (see 4.2.90), a Through_hole_condition (see 4.2.460), a Flat_with_radius_bottom (see 4.2.186), a Flat_bottom (see 4.2.183), a Spherical_bottom (see 4.2.421), or a Flat_with_chamfer_bottom (see 4.2.185).

The data associated with a Hole_bottom_condition are the following:

- is_applied_to.

The is_applied_to specifies to which end of the path the Hole_bottom_condition applies. The following values shall be used:

– 'start': The Hole_bottom_condition is applied to that end of the Round_hole (see 4.2.387) that corresponds to the start of the defining Linear_path (see 4.2.268);

– 'end': The Hole_bottom_condition is applied to that end of the Round_hole that corresponds to the end of the defining Linear_path.

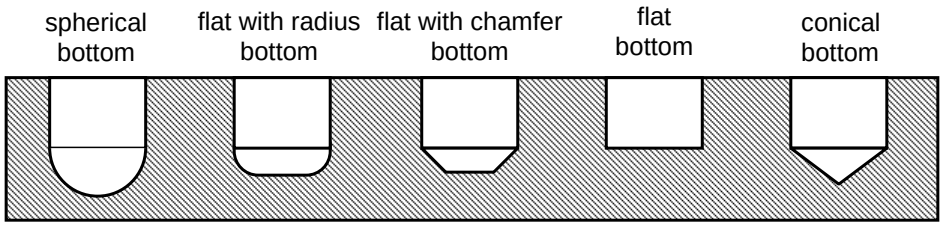


Figure 37 – Hole_bottom_condition

4.2.229 Hole_feature_in_panel

A Hole_feature_in_panel is a hole that is applied to a sheet like formed part.

A Hole_feature_in_panel is a type of Process_feature_in_panel (see 4.2.334).

Each Hole_feature_in_panel may be a Panel_hole_with_barring (see 4.2.299).

The data associated with a Hole_feature_in_panel are the following:

– hole_profile;

– hole_type;

– taper.

4.2.229.1 hole_profile

The hole_profile specifies a closed profile defining the shape of the Hole_feature_in_panel.

See 4.3.685 for the application assertion.

4.2.229.2 hole_type

The hole_type specifies the kind or shape of hole.

EXAMPLE 131 – 'circular' and 'rectangular' are examples for the hole_type.

The hole_type need not be specified for a particular Hole_feature_in_panel.

4.2.229.3 taper

The taper specifies a taper for the Hole_feature_in_panel.

See 4.3.684, 4.3.686, and 4.3.687 for the application assertions.

The taper need not be specified for a particular Hole_feature_in_panel.

If present, there shall be exactly one object that defines the taper for a Hole_feature_in_panel.

4.2.230 Hybrid_geometric_model_3d

A Hybrid_geometric_model_3d is a representation of geometry that consists of a set of Detailed_geometric_model_element (see 4.2.136) objects.

A Hybrid_geometric_model_3d is a type of Geometric_model (see 4.2.211).

The data associated with a Hybrid_geometric_model_3d are the following:

- element.

The element specifies a set of Detailed_geometric_model_element (see 4.2.136) that are part of the Hybrid_geometric_model_3d. In case this set is empty the Hybrid_geometric_model_3d shall reference at least one Template_instance (see 4.2.453) as 'additional_elements'.

See 4.3.688 for the application assertion.

4.2.231 Hyperbola

A Hyperbola is one branch of an open conic section defined by its radius, imaginary radius, the centre point and the direction of the line joining the centre to the focus.

A Hyperbola is a type of Curve (see 4.2.102).

4.2.232 Included_feature

An Included_feature is the identification of a feature as a component of a part. All geometric data defining the feature are defined in the part's coordinate space.

An Included_feature is a type of Feature_occurrence (see 4.2.176).

The data associated with an Included_feature are the following:

- definition.

The definition specifies the Feature_definition (see 4.2.175), the General_feature (see 4.2.196), the Transition_feature (see 4.2.476), or the Thread_feature (see 4.2.457) that has the information concerning the type of feature and parameters that describe the feature.

See 4.3.689, 4.3.690, 4.3.691, and 4.3.692 for the application assertions.

There shall be exactly one object that defines the definition for an Included_feature.

4.2.233 Initial_configuration

An Initial_configuration represents the initial values of the pair parameters of a Kinematic_mechanism (see 4.2.257). Some types of kinematic structures, e.g., four bar linkages, can not be uniquely only based on shape parameters.

EXAMPLE 132 – Link lengths are an example for shape parameters.

In those cases an Initial_configuration object provides pair parameter information that makes the configuration unique.

The data associated with an Initial_configuration are the following:

- initial_pair_values;
- is_applied_to.

4.2.233.1 initial_pair_values

The initial_pair_values specifies the initial value of the pair parameters of one or more Kinematic_pair (see 4.2.258) objects in a Kinematic_mechanism (see 4.2.257).

See 4.3.694 for the application assertion.

4.2.233.2 is_applied_to

The is_applied_to specifies the Kinematic_mechanism (see 4.2.257) that an Initial_configuration applies to.

See 4.3.693 for the application assertion.

4.2.234 Instance_placement

An Instance_placement is the information how to place a Single_instance (see 4.2.404) that is defined in its own Cartesian_coordinate_space (see 4.2.44) in the coordinate space of a superordinate Product_component (see 4.2.350).

The data associated with an Instance_placement are the following:

- placed_instance;
- product_component;
- transformation.

4.2.234.1 placed_instance

The placed_instance specifies the Single_instance (see 4.2.404) that is placed.

See 4.3.697 for the application assertion.

4.2.234.2 product_component

The product_component specifies the superordinate Product_component (see 4.2.350) that is defined in the reference coordinate space.

See 4.3.696 for the application assertion.

4.2.234.3 transformation

The transformation specifies the Geometric_model_relationship_with_transformation (see 4.2.213) or the Template_instance (see 4.2.453) that has the information about the geometric Transformation (see 4.2.472). In the case of Template_instance the scale for shall be omitted or set to 1.0.

See 4.3.695 and 4.3.698 for the application assertions.

There shall be exactly one object that defines the transformation for an Instance_placement.

4.2.235 Interpolated_configuration_sequence

An Interpolated_configuration_sequence is the motion (forward analysis) of a kinematic structure in terms of a set of configuration interpolations. These define the sequence of configuration definitions and the interpolation between them.

The data associated with an Interpolated_configuration_sequence are the following:

- interpolation.

The interpolation specifies the Kinematic_configuration_interpolation (see 4.2.253) for the Interpolated_configuration_sequence.

See 4.3.699 for the application assertion.

4.2.236 Item

An Item is either a single object or a unit in a group of objects. It collects the information that is common to all versions of the object.

NOTE – An Item may be either a single piece part, an assembly of arbitrary complexity, a raw material, or a tool.

EXAMPLE 133 – In the context of cars an Item may be the car as a whole, the assembly of the engine, the car body, a fender, the side window glass, grease, or stamping die.

The data associated with an Item are the following:

- description;
- id;
- name.

4.2.236.1 description

The description specifies additional information about the Item.

See 4.3.700 for the application assertion.

The description need not be specified for a particular Item.

If present, there shall be exactly one object that defines the description for an Item.

4.2.236.2 id

The id specifies the identifier of the Item. This identifier shall be unique.

NOTE – The scope of uniqueness is usually dependent on the form of implementation; it may be a physical file or data base.

4.2.236.3 name

The name specifies the word or group of words used to refer to the Item.

EXAMPLE 134 – 'Ford T', 'Fender 4XA', or '2CV front bumper' are examples for name.

See 4.3.700 for the application assertion.

There shall be exactly one object that defines the name for an Item.

4.2.237 Item_classification

An Item_classification is a classification of an Item (see 4.2.236) with respect to specific criteria.

NOTE – If an Item requires classification by more than one criterion, several Item_classification objects are associated to the same Item.

Each Item_classification is either a General_classification (see 4.2.192) or a Specific_classification (see 4.2.414).

The data associated with an Item_classification are the following:

- associated_item;
- description.

4.2.237.1 associated_item

The associated_item specifies the Item (see 4.2.236) with which a particular Item_classification (see 4.2.237) is associated.

NOTE – The set of associated items may be empty in the case of a high level classification that is further decomposed in a hierarchy.

See 4.3.701 for the application assertion.

4.2.237.2 description

The description specifies additional information about the Item_classification.

See 4.3.702 for the application assertion.

The description need not be specified for a particular Item_classification.

If present, there shall be exactly one object that defines the description for an Item_classification.

4.2.238 Item_cost_property

An Item_cost_property is an item related property that specifies the costs that are connected with it.

An Item_cost_property is a type of Item_property (see 4.2.243).

The data associated with an Item_cost_property are the following:

- cost_property_value.

The cost_property_value specifies the value of the Item_cost_property.

NOTE – The interpretation of this value, e.g., whether the value is required or calculated, is dependent on the semantics of the attribute 'value_interpretation' of the referenced Property_value.

See 4.3.703 for the application assertion.

4.2.239 Item_definition_instance_relationship

An Item_definition_instance_relationship is a relationship between a Design_discipline_item_definition (see 4.2.134) and an Item_instance (see 4.2.241).

NOTE – This relationship does not comprise inheritance of data from the relating Design_discipline_item_definition to the related Item_instance.

Each Item_definition_instance_relationship is either a Collected_item_association (see 4.2.76), a Mated_item_association (see 4.2.275), an Assembly_component_relationship (see 4.2.26), or a General_item_definition_instance_relationship (see 4.2.198).

The data associated with an Item_definition_instance_relationship are the following:

- related;
- relating.

4.2.239.1 related

The related specifies the Item_instance (see 4.2.241) that is part of the Item_definition_instance_relationship.

NOTE – The semantic of this attribute is defined by the specializations of Item_definition_instance_relationship.

See 4.3.705 for the application assertion.

4.2.239.2 relating

The relating specifies the Design_discipline_item_definition (see 4.2.134) that is part of the Item_definition_instance_relationship.

NOTE – The semantic of this attribute is defined by the specializations of Item_definition_instance_relationship.

See 4.3.704 for the application assertion.

4.2.240 Item_definition_relationship

An Item_definition_relationship is a relationship between two Design_discipline_item_definition (see 4.2.134) objects.

NOTE – This relationship does not comprise inheritance of data from the relating Design_discipline_item_definition to the related Design_discipline_item_definition.

Each Item_definition_relationship is either a Tool_part_relationship (see 4.2.467), a Replaced_definition_relationship (see 4.2.377), a Geometrical_relationship (see 4.2.219), a Make_from_relationship (see 4.2.273), or a General_item_definition_relationship (see 4.2.199).

The data associated with an Item_definition_relationship are the following:

- related;
- relating.

4.2.240.1 related

The related specifies the second of the Design_discipline_item_definition (see 4.2.134) objects that are part of the relationship.

NOTE – The semantic of this attribute is defined by the specializations of Item_definition_relationship.

See 4.3.706 for the application assertion.

4.2.240.2 relating

The relating specifies the first of the Design_discipline_item_definition (see 4.2.134) objects that are part of the relationship.

NOTE – The semantic of this attribute is defined by the specializations of Item_definition_relationship.

See 4.3.706 for the application assertion.

4.2.241 Item_instance

An Item_instance is the occurrence of an object that is defined either as a Design_discipline_item_definition (see 4.2.134) or as a Product_specification (see 4.2.354). An Item_instance shall be used at least once as an Item_solution (see 4.2.246) or in an Assembly_component_relationship (see 4.2.26).

EXAMPLE 135 – In the case of a car the Item (see 4.2.236) 'wheel' is defined once. This Design_discipline_item_definition (see 4.2.134) carries all the information defining the wheel (e.g. its shape) that is independent from its usage. Additionally there are 5 Item_instance objects for this wheel since there are 5 equal wheels used in a car: Front-left, front-right, rear-left, rear-right, and spare wheel. Each of these instances may carry additional information like placement.

Each Item_instance is either a Single_instance (see 4.2.404), a Quantified_instance (see 4.2.365), or a Selected_instance (see 4.2.392).

The data associated with an Item_instance are the following:

- definition;
- description;
- id.

4.2.241.1 definition

The definition specifies the Design_discipline_item_definition (see 4.2.134) or the Product_specification (see 4.2.354) that serves as a definition for this particular occurrence.

See 4.3.707 and 4.3.709 for the application assertions.

There shall be exactly one object that defines the definition for an Item_instance.

4.2.241.2 description

The description specifies additional information about the Item_instance.

See 4.3.708 for the application assertion.

The description need not be specified for a particular Item_instance.

If present, there shall be exactly one object that defines the description for an Item_instance.

4.2.241.3 id

The id specifies the identifier of the Item_instance.

4.2.242 Item_instance_relationship

An Item_instance_relationship is a relationship between two Item_instance (see 4.2.241) objects.

NOTE – This relationship does not comprise inheritance of data from the relating Item_instance to the related Item_instance.

Each Item_instance_relationship is either a General_item_instance_relationship (see 4.2.200), a Same_time_machining_relationship (see 4.2.389), or a Replaced_usage_relationship (see 4.2.378).

The data associated with an Item_instance_relationship are the following:

- related;
- relating.

4.2.242.1 related

The related specifies the second of the two objects related by the Item_instance_relationship.

NOTE – The semantic of this attribute is defined by the specializations of Item_instance_relationship.

See 4.3.710 for the application assertion.

4.2.242.2 relating

The relating specifies the first of the two objects related by the Item_instance_relationship.

NOTE – The semantic of this attribute is defined by the specializations of Item_instance_relationship.

See 4.3.710 for the application assertion.

4.2.243 Item_property

An Item_property is a characteristic of an item.

An Item_property is a type of Property (see 4.2.361).

Each Item_property is either a General_item_property (see 4.2.201), a Recyclability (see 4.2.375), an Item_cost_property (see 4.2.238), a Mass (see 4.2.274), a Material_property (see 4.2.279), or an Item_quality_property (see 4.2.244).

The data associated with an Item_property are the following:

- described_object.

The described_object specifies the object that the Item_property is associated to.

See 4.3.711, 4.3.712, 4.3.713, 4.3.714, 4.3.715, 4.3.716, 4.3.717, 4.3.718, 4.3.719, 4.3.720, 4.3.721, 4.3.722, 4.3.723, 4.3.724, 4.3.725, 4.3.726, 4.3.727, and 4.3.728 for the application assertions.

There shall be exactly one object that defines the described_object for an Item_property.

4.2.244 Item_quality_property

An Item_quality_property is a property that documents the results of any quality measure process.

EXAMPLE 136 – Within prototype building several alternate manufacturing technologies are tested. An Item_quality_property for the resulting parts, used tools, and the used process operations may document the level of reached quality for the parts.

An Item_quality_property is a type of Item_property (see 4.2.243).

The data associated with an Item_quality_property are the following:

- description;
- quality_property_value.

4.2.244.1 description

The description specifies additional information about the Item_quality_property.

See 4.3.729 for the application assertion.

The description need not be specified for a particular Item_quality_property.

If present, there shall be exactly one object that defines the description for an Item_quality_property.

4.2.244.2 quality_property_value

The quality_property_value specifies the value of the Item_quality_property.

NOTE – The interpretation of this value, e.g., whether the value is required or calculated, is dependent on the semantics of the attribute 'value_interpretation' of the referenced Property_value.

See 4.3.730 for the application assertion.

4.2.245 Item_shape

An Item_shape is the definition of the shape of an Item_version (see 4.2.248).

Each Item_shape may be a Featured_shape (see 4.2.179).

The data associated with an Item_shape are the following:

- described_object;
- description.

4.2.245.1 described_object

The described_object specifies the object that the Item_shape defines the shape for.

See 4.3.731, 4.3.732, 4.3.733, 4.3.734, and 4.3.735 for the application assertions.

There shall be exactly one object that defines the described_object for an Item_shape.

4.2.245.2 description

The description specifies additional information about the Item_shape.

See 4.3.736 for the application assertion.

The description need not be specified for a particular Item_shape.

If present, there shall be exactly one object that defines the description for an Item_shape.

4.2.246 Item_solution

An Item_solution is a solution to a functional requirement. It specifies a set of items that jointly fulfill the same functional requirements. The Item_solution objects for the same base represent the mutually exclusive alternatives. In the case where the attribute 'base element' refers to another Item_solution there shall not be an instance of Item_solution_relationship (see 4.2.247) relating these two Item_solution objects.

NOTE – An Item_solution may refer directly to the functional requirement that is represented as a Product_component (see 4.2.350) or to another Item_solution in which case it serves as a refinement of that Item_solution.

Each Item_solution is either a Technical_solution (see 4.2.451), a Final_item_solution (see 4.2.182), or a Supplier_solution (see 4.2.433).

The data associated with an Item_solution are the following:

- base_element;
- id;
- version_id.

4.2.246.1 base_element

The base_element specifies the object the Item_solution provides solutions for.

See 4.3.737 and 4.3.738 for the application assertions.

There shall be exactly one object that defines the base_element for an Item_solution.

4.2.246.2 id

The id specifies the identifier of the Item_solution.

4.2.246.3 version_id

The version_id specifies the identification of a particular version of an Item_solution.

The version_id need not be specified for a particular Item_solution.

4.2.247 Item_solution_relationship

An Item_solution_relationship is a relationship between two Item_solution (see 4.2.246) objects. The related Item_solution shall not refer to the relating Item_solution as 'base_element'.

The data associated with an Item_solution_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.247.1 description

The description specifies additional information about the Item_solution_relationship.

See 4.3.740 for the application assertion.

The description need not be specified for a particular Item_solution_relationship.

If present, there shall be exactly one object that defines the description for an Item_solution_relationship.

4.2.247.2 related

The related specifies the second of the two Item_solution (see 4.2.246) objects related by the Item_solution_relationship.

See 4.3.739 for the application assertion.

4.2.247.3 relating

The relating specifies the first of the two Item_solution (see 4.2.246) objects related by the Item_solution_relationship.

See 4.3.739 for the application assertion.

4.2.247.4 relation_type

The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:

- 'derivation': The application object defines a relationship where the related Item_solution is derived from the relating Item_solution;

- 'sequence': This type of Item_solution_relationship defines a version sequence where the relating Item_solution (see 4.2.246) is the preceding version and the related Item_solution is the following version.

EXAMPLE 137 – The relation_type 'derivation' may be used if the relating Item_solution serves as the base for the development of the related Item_solution (e.g. in the case of process planning).

4.2.248 Item_version

An Item_version is the identification of a physically realizable object. It collects the information defining this object.

NOTE 1 – An Item_version may be produced, consumed, used to produce other Item_version objects, or offered to the market.

NOTE 2 – The collection of defining information may be incomplete, i.e. not all of the defining Design_discipline_item_definition (see 4.2.134) objects are associated with an Item_version (see 4.2.248).

NOTE 3 – The set of Item_version objects of an Item represents the history of an Item within a particular life cycle stage or over the whole life cycle.

The data associated with an Item_version are the following:

- associated_item;

- description;

- id.

4.2.248.1 associated_item

The associated_item specifies the Item (see 4.2.236) with which the Item_version is associated.

See 4.3.741 for the application assertion.

4.2.248.2 description

The description specifies additional information about the Item_version.

See 4.3.742 for the application assertion.

The description need not be specified for a particular Item_version.

If present, there shall be exactly one object that defines the description for an Item_version.

4.2.248.3 id

The id specifies the identifier of the Item_version. The id shall be unique within the scope of the associated Item (see 4.2.236).

NOTE – A particular object is identified by the id of Item_version that serves to identify the particular version coupled with the id of the associated Item that serves to identify the generic object.

4.2.249 Item_version_relationship

An Item_version_relationship is a relationship between two Item_version (see 4.2.248) objects.

NOTE – There is no inheritance of data between the two related Item_version objects.

The data associated with an Item_version_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.249.1 description

The description specifies additional information about the Item_version_relationship.

See 4.3.744 for the application assertion.

The description need not be specified for a particular Item_version_relationship.

If present, there shall be exactly one object that defines the description for an Item_version_relationship.

4.2.249.2 related

The related specifies the second of the two Item_version (see 4.2.248) objects related by the Item_version_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.743 for the application assertion.

4.2.249.3 relating

The relating specifies the the first of the two Item_version (see 4.2.248) objects related by the Item_version_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.743 for the application assertion.

4.2.249.4 relation_type

The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:

– 'hierarchy': The application object defines a hierarchical relationship where the related Item_version is a sub version of the relating Item_version (e.g. revision 1.1 and 1.2);

– 'sequence': The application object defines a version sequence where the relating Item_version is the preceding version and the related Item_version is the following version. For an Item_version there shall be at the most one Item_version_relationship of this relation_type as relating and at most one Item_version_relationship of this relation_type as related;

– 'derivation': The application object defines a deriving relationship where the related Item_version is based on the relating Item_version which is an earlier version of the same or of a different Item (see 4.2.236);

– 'alias': The application object defines a relationship where the related Item_version is an alias for the relating Item_version. In this case the Item associated with the two Item_version objects shall be different.

EXAMPLE 138 – Usually items that are provided by suppliers have a different id at the supplier's site than on the manufacturer's site: an item that is identified in a company by the identifier 'C425' and the version id 'V2.0' may correspond to an item the supplier identifies as '2X45' with the version id 'V3.2'.

NOTE – For Item_version the concept of Alias_identification (see 4.2.7) is not used. Instead the Item_version_relationship with relation type 'alias' is used. In the case of Item_version there is other information besides id and version id, such as the name or attached approval information, that may also be different.

4.2.250 Joggle_feature

A Joggle_feature is a change in elevation of a face along a rim designed to accommodate either clearance for the insertion of another part or a contact point for attachment of another part.

EXAMPLE 139 – Figure 38 illustrates the meaning of the various parameters defining a Joggle_-feature.

A Joggle_feature is a type of Process_feature_in_panel (see 4.2.334).

The data associated with a Joggle_feature are the following:

- defining_path;
- direction_angle_at_path_end;
- direction_angle_at_path_start;
- height;
- slope_angle_at_path_end;
- slope_angle_at_path_start;
- termination.

4.2.250.1 defining_path

The defining_path specifies the Linear_path (see 4.2.268) whose start and end point in combination with the angles at path start and path end define the location and orientation of the side walls of the Joggle_feature. The Linear_path shall be defined in the same Cartesian_coordinate_-space (see 4.2.44) as the Joggle_feature and shall be coplanar with the panel.

See 4.3.747 for the application assertion.

4.2.250.2 direction_angle_at_path_end

The direction_angle_at_path_end specifies an angle between 0 degrees and 90 degrees that is measured between the Linear_path (see 4.2.268) and a line starting at the end point of the Linear_path and being directed towards the rim of the panel that shall have the Joggle_feature. This line shall be coplanar with the panel, with the Linear_path and with one side wall of the Joggle_feature.

See 4.3.745 for the application assertion.

4.2.250.3 direction_angle_at_path_start

The direction_angle_at_path_start specifies an angle between 0 degrees and 90 degrees that is measured between the Linear_path (see 4.2.268) and a line starting at the start point of the

Linear_path and being directed towards the rim of the panel that shall have the Joggle_feature. This line shall be coplanar with the panel, with the Linear_path and with one side wall of the Joggle_feature.

See 4.3.745 for the application assertion.

4.2.250.4 height

The height specifies the distance between the panel surface and the top face of the Joggle_feature.

See 4.3.745 for the application assertion.

4.2.250.5 slope_angle_at_path_end

The slope_angle_at_path_end specifies the slope angle of that side wall of the Joggle_feature that is defined by the line starting at the end of the Linear_path (see 4.2.268).

See 4.3.745 for the application assertion.

4.2.250.6 slope_angle_at_path_start

The slope_angle_at_path_start specifies the slope angle of that side wall of the Joggle_feature that is defined by the line starting at the start of the Linear_path (see 4.2.268).

See 4.3.745 for the application assertion.

4.2.250.7 termination

The termination specifies the Joggle_termination (see 4.2.251) that defines the Joggle_feature's end condition.

See 4.3.746 for the application assertion.

4.2.251 Joggle_termination

A Joggle_termination is the end condition specifying the termination of a Joggle_feature (see 4.2.250).

Each Joggle_termination is either a Face_termination (see 4.2.169) or a Value_termination (see 4.2.484).

4.2.252 Kinematic_configuration_definition

A Kinematic_configuration_definition is a configuration of a kinematic structure in terms of pair parameter sets.

The data associated with a Kinematic_configuration_definition are the following:

- pair_values;

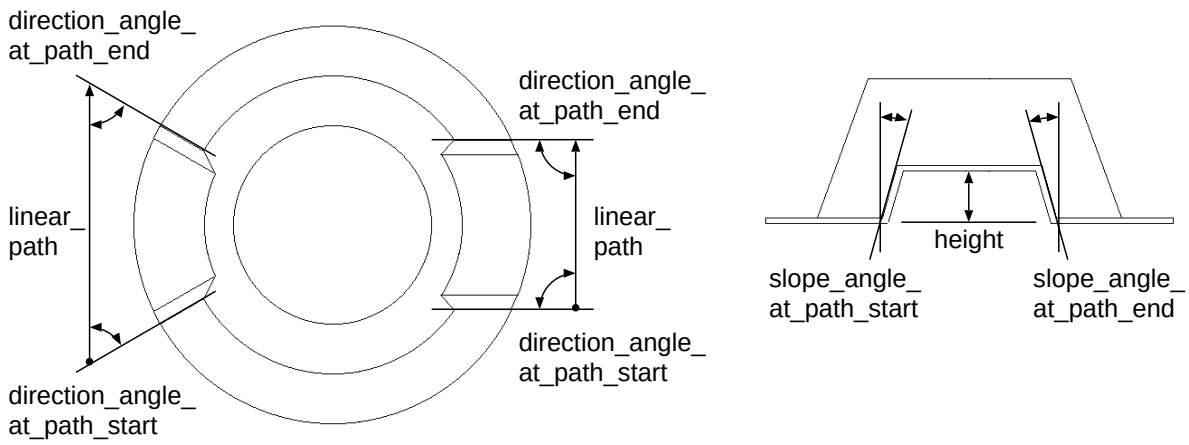


Figure 38 – Parameters defining a Joggle_feature

– t_parameter.

4.2.252.1 pair_values

The pair_values specifies the set of values applied to the pairs that specify a Kinematic_configuration_definition.

See 4.3.748 for the application assertion.

4.2.252.2 t_parameter

The t_parameter specifies the parameter associated to the Kinematic_configuration_definition.

See 4.3.749 for the application assertion.

4.2.253 Kinematic_configuration_interpolation

A Kinematic_configuration_interpolation is the specification of the interpolation of the configurations between two configuration definitions.

The data associated with a Kinematic_configuration_interpolation are the following:

– interpolation_type;

– next_definition;

– previous_definition.

4.2.253.1 interpolation_type

The interpolation_type specifies the type of interpolation from the previous configuration definition to the next.

4.2.253.2 next_definition

The next_definition specifies the Kinematic_configuration_definition (see 4.2.252) at which the interpolation ends.

See 4.3.750 for the application assertion.

4.2.253.3 previous_definition

The previous_definition specifies the Kinematic_configuration_definition (see 4.2.252) from which the interpolation starts.

See 4.3.750 for the application assertion.

4.2.254 Kinematic_joint

A Kinematic_joint is the topological aspect of an ordered connection between exactly two links. The orientation is always implicitly associated with a joint. This orientation points from the first link to the second link.

EXAMPLE 140 – In case of a rack and pinion pair the orientation of the Kinematic_joint would reflect the transformation to a link representing the pinion.

A Kinematic_joint is a type of Detailed_model_element (see 4.2.137).

The data associated with a Kinematic_joint are the following:

- first_link;
- pair;
- second_link.

4.2.254.1 first_link

The first_link specifies the first link of the ordered connection at the Kinematic_joint.

See 4.3.751 for the application assertion.

4.2.254.2 pair

The pair specifies the geometric constraints of the Kinematic_joint (see 4.2.254).

See 4.3.752 for the application assertion.

4.2.254.3 second_link

The second_link specifies the second link of the ordered connection at the Kinematic_joint.

See 4.3.751 for the application assertion.

4.2.255 Kinematic_link

A Kinematic_link is the topological information associated with a rigid part of a Kinematic_mechanism (see 4.2.257).

A Kinematic_link is a type of Detailed_model_element (see 4.2.137).

The data associated with a Kinematic_link are the following:

- representation.

The representation specifies the geometrical representation and the implicit link frame of the Kinematic_link.

See 4.3.753 for the application assertion.

4.2.256 Kinematic_link_representation

A Kinematic_link_representation is the description of the geometry of a Kinematic_link (see 4.2.255).

The data associated with a Kinematic_link_representation are the following:

- additional_reference_frame;
- physical_description.

4.2.256.1 additional_reference_frame

The additional_reference_frame specifies additional frames that are attached to the Kinematic_link_representation.

See 4.3.755 for the application assertion.

4.2.256.2 physical_description

The physical_description specifies the shape of the Kinematic_link_representation (see 4.2.256). The physical_description may also define the mass properties of the Kinematic_link_representation.

See 4.3.754 for the application assertion.

4.2.257 Kinematic_mechanism

A Kinematic_mechanism is a Kinematic_structure (see 4.2.259) and the position of this Kinematic_structure relative to the reference coordinate system of the parent object. A Kinematic_-

mechanism describes a mechanical product whose motion is constrained by joints.

The data associated with a Kinematic_mechanism are the following:

- base;
- base_placement;
- structure_definition.

4.2.257.1 base

The base specifies the Kinematic_link (see 4.2.255) relative to that all the other Kinematic_link objects may move. The base may be a part of another Kinematic_mechanism.

EXAMPLE 141 – A gripping tool attached to the end of a robot arm is an example for base.

This does not mean that Kinematic_mechanism objects are limited to those which have the same base at all times.

EXAMPLE 142 – In space applications, e.g., worm-like mechanisms have been designed that grab to a space-structure with one end, move on with the other end, fix themselves with an end-effector, and then loosen the original grip.

See 4.3.756 for the application assertion.

4.2.257.2 base_placement

The base_placement describes the rigid placement of the link frame of the base relative to the reference coordinate system of the parent object of the mechanism.

See 4.3.758 for the application assertion.

4.2.257.3 structure_definition

The structure_definition specifies the Kinematic_structure (see 4.2.259) that makes up the Kinematic_mechanism.

See 4.3.757 for the application assertion.

4.2.258 Kinematic_pair

A Kinematic_pair represents the kinematic restriction between two adjacent Kinematic_link (see 4.2.255) objects coinciding in a Kinematic_joint (see 4.2.254).

EXAMPLE 143 – Figure 39 shows an example for placement relationships between kinematic pair frames and mating links.

The data associated with a Kinematic_pair are the following:

- transformation_item_1;
- transformation_item_2.

4.2.258.1 transformation_item_1

The transformation_item_1 specifies the rigid placement of the pair frame relative to the link frame for the first Kinematic_link (see 4.2.255) referenced by the Kinematic_joint (see 4.2.254) associated with the Kinematic_pair.

See 4.3.759 for the application assertion.

4.2.258.2 transformation_item_2

The transformation_item_2 specifies the rigid placement of the pair frame relative to the link frame for the second Kinematic_link (see 4.2.255) referenced by the Kinematic_joint (see 4.2.254) associated with the Kinematic_pair.

See 4.3.759 for the application assertion.

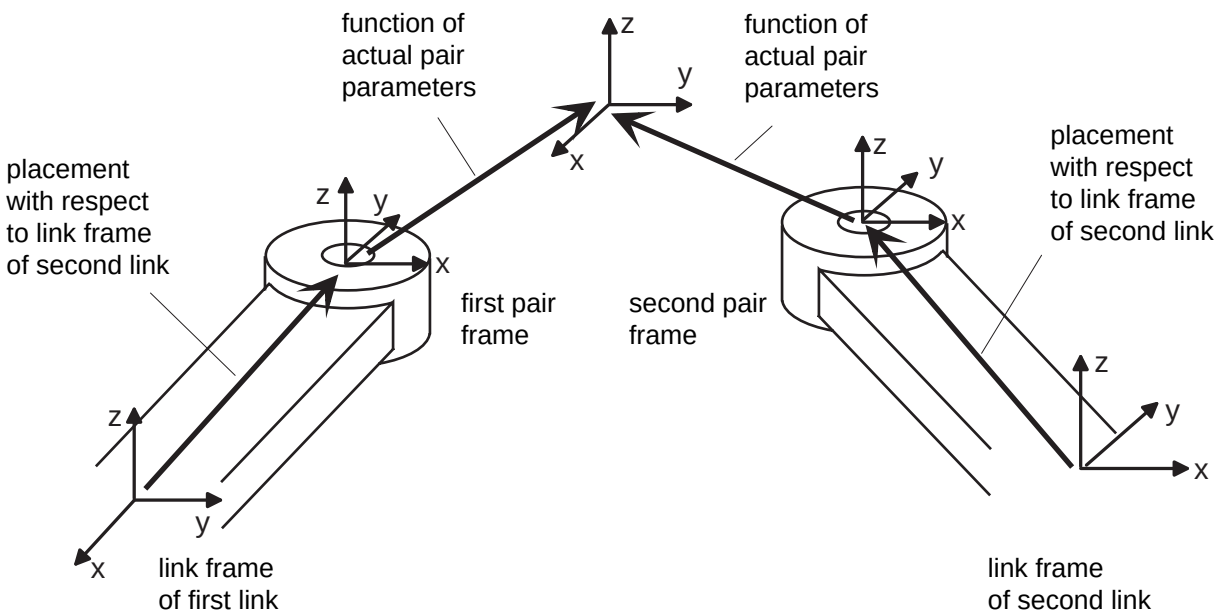


Figure 39 – Placement relationships kinematic pair frames and mating links

4.2.259 Kinematic_structure

A Kinematic_structure specifies the topological aspects of a kinematic representation. The topology of the Kinematic_structure is represented by means of kinematic joints, which in turn, establish a relationship between pairs of kinematic links.

A Kinematic_structure is a type of Detailed_model_element (see 4.2.137).

The data associated with a Kinematic_structure are the following:

- joint.

The joint specifies the Kinematic_joint (see 4.2.254) that is part of the Kinematic_structure (see 4.2.259).

See 4.3.760 for the application assertion.

4.2.260 Kinematic_transformation_result

A Kinematic_transformation_result is the result information of a kinematic simulation of a kinematic mechanism.

The data associated with a Kinematic_transformation_result are the following:

- is_applied_to;
- result.

4.2.260.1 is_applied_to

The is_applied_to specifies the mechanism whose motion produced the results.

See 4.3.762 for the application assertion.

4.2.260.2 result

The result specifies the result information of the Kinematic_transformation_result that is an interpolated configuration sequence defining the motion of a kinematic structure through interpolations.

See 4.3.761 for the application assertion.

4.2.261 Layer

A Layer is a general structure for the collection of geometric and annotation elements. Layers shall not be nested, i.e., a layer cannot be put on another layer.

EXAMPLE 144 – The design of a mechanical part as a B_rep_model (see 4.2.35) is performed in several steps:

- creation of wireframe elements;
- use the wireframe elements to design the surfaces;
- use the surfaces to design the B_rep_model.

The elements created in each stage are put on separate layers. After the creation of the B_rep_model the wireframe elements can be looked at as 'auxiliary geometry' that becomes the purpose of its layer. The visibility control of the layer is usually used to display only the B_rep_model or to display the auxiliary wireframe elements in addition, e.g., for design changes.

The data associated with a Layer are the following:

- description;
- element;
- id.

4.2.261.1 description

The description specifies additional information about the Layer.

See 4.3.764 for the application assertion.

The description need not be specified for a particular Layer.

If present, there shall be exactly one object that defines the description for a Layer.

4.2.261.2 element

The element specifies a piece of product data that is collected by a Layer.

NOTE – Although not explicitly modelled in the ARM, the element can be on various levels of granularity, e.g., a whole B-rep-model or a single Cartesian point.

See 4.3.763 for the application assertion.

4.2.261.3 id

The id specifies the identifier of the Layer.

4.2.262 Leader

A Leader directs a dimension, a note, or a symbol to the intended place or point on a geometric or annotation element that is part of a Drawing_sheet (see 4.2.152), a View (see 4.2.490), a View_area (see 4.2.491), or of a Geometric_model (see 4.2.211).

A Leader is a type of Annotation_curve (see 4.2.14).

Each Leader may be an Associated_leader (see 4.2.31).

The data associated with a Leader are the following:

- terminator.

The terminator specifies an annotation symbol that is applied to a leader and used to identify the endpoint or point of application of the directed annotation.

See 4.3.765 for the application assertion.

The terminator need not be specified for a particular Leader.

4.2.263 Light_source

A Light_source is a luminous source that affects the display of surfaces of three-dimensional geometric models within a defined Model_image (see 4.2.284).

The data associated with a Light_source are the following:

- colour;
- light_type;
- view.

4.2.263.1 colour

The colour specifies the colour of a light source that is defined by the intensities of red, green, and blue proportions. The colour is either userdefined or predefined. Figure 19 shows the predefined colours.

The colour need not be specified for a particular Light_source.

4.2.263.2 light_type

The light_type specifies the kind of Light_source, such as directional, positional, spot light, or ambient. Position and attenuation as well as additional characteristics of the Light_source are specified dependent on the kind of Light_source. The position of the Light_source, if needed to be specified, is defined in the Cartesian coordinate space of the shape model.

4.2.263.3 view

The view specifies the Model_image (see 4.2.284) that the Light_source is applied to.

See 4.3.766 for the application assertion.

4.2.264 Limits_and_fits

A Limits_and_fits contains the necessary information to express a tolerance of the limits-and-fits system standardized by ISO 286.

The data associated with a Limits_and_fits are the following:

- deviation;
- fitting_type;
- grade.

4.2.264.1 deviation

The deviation specifies the enumeration of limits and fits class descriptors by characters.

NOTE – The characters 'A' to 'CZ' for holes or 'a' to 'cz' for shafts according to the ISO 286 system 'limits-and-fits' may be used for deviation.

4.2.264.2 fitting_type

The fitting_type specifies whether a tolerance declaration within the limits and fits system relates to a shaft or a hole.

The fitting_type need not be specified for a particular Limits_and_fits.

4.2.264.3 grade

The grade specifies the quality or accuracy grade of a tolerance.

NOTE – The grade may be based on the international standard tolerance grade IT01 to IT18.

4.2.265 Line

A Line is an unbounded curve that is defined by a point and a vector. The positive direction of the line is in the direction of the vector and the magnitude of the vector controls the parametrization.

A Line is a type of Curve (see 4.2.102).

4.2.266 Linear_bead_end_type

A Linear_bead_end_type is an end condition of a bead feature that shall follow a linear path out of the material.

A Linear_bead_end_type is a type of Bead_end_type (see 4.2.38).

The data associated with a Linear_bead_end_type are the following:

- end_length;
- end_shape.

4.2.266.1 end_length

The end_length specifies the length of the Linear_bead_end_type.

4.2.266.2 end_shape

The end_shape specifies if the linear bead end is straight on the panel surface or tapered towards the end point.

4.2.267 Linear_distance_dimension

A Linear_distance_dimension is a dimension where the distance is calculated between two elements along a linear path.

A Linear_distance_dimension is a type of Location_dimension (see 4.2.270).

The data associated with a Linear_distance_dimension are the following:

- directed;
- orientation.

4.2.267.1 directed

The directed specifies a logical value designating the importance of direction for measuring a Linear_distance_dimension. If the value is 'true' the Linear_distance_dimension is measured from origin to target. A value of 'false' indicates that the direction for measuring Linear_distance_dimension is inconsequential.

4.2.267.2 orientation

The orientation specifies a point and a direction along which the Linear_distance_dimension is measured.

4.2.268 Linear_path

A Linear_path is a direction of travel along a line.

EXAMPLE 145 – Figure 40 illustrates a Linear_path.

A Linear_path is a type of Path (see 4.2.304).

The data associated with a Linear_path are the following:

- distance;
- is_defined_by.

4.2.268.1 distance

The distance specifies the length of the Linear_path.

See 4.3.768 for the application assertion.

4.2.268.2 is_defined_by

The is_defined_by specifies the Axis_placement (see 4.2.33) that defines the location of the start point and the direction of the Linear_path. The start point is the origin of the Axis_placement and the direction is the z-axis of the Axis_placement.

The Axis_placement is defined in the same Cartesian_coordinate_space (see 4.2.44) as the Linear_path. In the case where the Linear_path is used to define the sweep path of a profile the x-y-plane of the Axis_placement shall coincide with the x-y-plane of the profile.

See 4.3.767 for the application assertion.

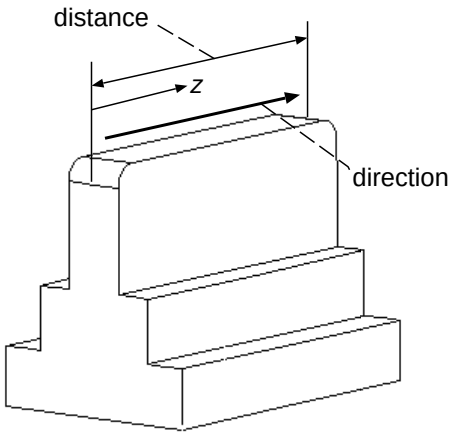


Figure 40 – Linear_path

4.2.269 Linear_profile_tolerance

A Linear_profile_tolerance is the definition of a uniform boundary along the true profile within which the elements of the surface must lie. The tolerance zone established by a Linear_profile_tolerance is two dimensional, extending along the length of the considered feature as a whole respectively along each segment of a defined size if the segment attribute is specified.

A Linear_profile_tolerance is a type of Geometric_tolerance (see 4.2.216).

The data associated with a Linear_profile_tolerance are the following:

- affected_plane;
- datum.

4.2.269.1 affected_plane

The affected_plane specifies the plane a certain type of geometric tolerance applies in. This attribute covers information that is given on drawings implicitly by the drawing plane. If the affected_plane attribute is not specified the tolerance applies in all directions which results in a cylindrical tolerance zone (on drawings indicated by the diameter symbol).

See 4.3.770 for the application assertion.

The affected_plane need not be specified for a particular Linear_profile_tolerance.

4.2.269.2 datum

The datum specifies the Datum_feature (see 4.2.117) or the Compound_datum (see 4.2.80) that defines the reference for this type of geometric tolerance.

See 4.3.769 for the application assertion.

4.2.270 Location_dimension

A Location_dimension is a dimension characterized as the distance between two elements of geometry or annotation (origin and target).

A Location_dimension is a type of Geometric_dimension (see 4.2.210).

Each Location_dimension is either a Curved_distance_dimension (see 4.2.107), a Linear_distance_dimension (see 4.2.267), or an Angular_location_dimension (see 4.2.11).

The data associated with a Location_dimension are the following:

- origin;
- target.

4.2.270.1 origin

The origin specifies the geometric or annotation element which the Location_dimension is associated with as origin.

See 4.3.771, 4.3.772, 4.3.773, and 4.3.774 for the application assertions.

There shall be exactly one object that defines the origin for a Location_dimension.

4.2.270.2 target

The target specifies the geometric or annotation element which the Location_dimension is associated with as target.

See 4.3.771, 4.3.772, 4.3.773, and 4.3.774 for the application assertions.

There shall be exactly one object that defines the target for a Location_dimension.

4.2.271 Locator_feature

A Locator_feature is a conical impression in a panel used to locate the panel in the tooling.

NOTE – Locator_feature objects are usually created in early die strikes for further stamping.

A Locator_feature is a type of Process_feature_in_panel (see 4.2.334).

The data associated with a Locator_feature are the following:

- diameter;
- tip_angle;

- tip_radius.

4.2.271.1 diameter

The diameter specifies the diameter of the cone on the panel's surface.

See 4.3.775 for the application assertion.

4.2.271.2 tip_angle

The tip_angle specifies the opening angle of the cone.

See 4.3.775 for the application assertion.

4.2.271.3 tip_radius

The tip_radius specifies the radius of the rounded cone tip.

See 4.3.775 for the application assertion.

4.2.272 Lot_configuration

A Lot_configuration is a Configuration (see 4.2.89) that applies for a given production batch of a product.

A Lot_configuration is a type of Configuration (see 4.2.89).

The data associated with a Lot_configuration are the following:

- lot_id;
- lot_size.

4.2.272.1 lot_id

The lot_id specifies the identification of the batch for which the Lot_configuration applies.

4.2.272.2 lot_size

The lot_size specifies the size of the batch for which the Lot_configuration applies.

4.2.273 Make_from_relationship

A Make_from_relationship is a relationship between a Design_discipline_item_definition (see 4.2.134) defining raw material or semi finished parts (related) and a Design_discipline_item_definition defining the Item_version (see 4.2.248) manufactured out of that (relating).

NOTE – The inherited attribute related specifies the raw material or semi finished item, and the inherited attribute relating specifies the manufactured item.

A Make_from_relationship is a type of Item_definition_relationship (see 4.2.240).

The data associated with a Make_from_relationship are the following:

- description;
- name.

4.2.273.1 description

The description specifies additional information about the Make_from_relationship.

See 4.3.776 for the application assertion.

The description need not be specified for a particular Make_from_relationship.

If present, there shall be exactly one object that defines the description for a Make_from_relationship.

4.2.273.2 name

The name specifies the word or group of words by which the Make_from_relationship is referred to.

See 4.3.776 for the application assertion.

The name need not be specified for a particular Make_from_relationship.

If present, there shall be exactly one object that defines the name for a Make_from_relationship.

4.2.274 Mass

A Mass is a quantity of matter that an item contains.

A Mass is a type of Item_property (see 4.2.243).

The data associated with a Mass are the following:

- mass_property_value.

The mass_property_value specifies the value of the Mass.

NOTE – The interpretation of this value, e.g., whether the value is required or calculated, is dependent on the semantics of the attribute 'value_interpretation' of the referenced Property_value.

See 4.3.777 for the application assertion.

4.2.275 Mated_item_association

A Mated_item_association is a relationship between a Mating_definition (see 4.2.282) and an Item_instance (see 4.2.241) that is used within the Mating_definition.

A Mated_item_association is a type of Item_definition_instance_relationship (see 4.2.239).

The data associated with a Mated_item_association are the following:

- relating.

The relating specifies the Mating_definition (see 4.2.282) that is the result of the mating operation.

See 4.3.778 for the application assertion.

4.2.276 Mated_item_relationship

A Mated_item_relationship is a relationship between two Mated_item_association (see 4.2.275) objects. This relationship specifies additional information about the mating of two particular items that go into a Mating_definition (see 4.2.282). The two Mated_item_association objects that are referenced by the Mated_item_relationship shall refer to the same Mating_definition.

The data associated with a Mated_item_relationship are the following:

- mated_shape;
- mating_material;
- related;
- relating;
- transformation.

4.2.276.1 mated_shape

The mated_shape specifies the Shape_aspect_relationship (see 4.2.395) that relates the two Shape_aspect (see 4.2.394) objects that form the area of mating contact.

See 4.3.782 for the application assertion.

The mated_shape need not be specified for a particular Mated_item_relationship.

4.2.276.2 mating_material

The mating_material specifies the Material (see 4.2.277) used for the mounting.

See 4.3.781 for the application assertion.

4.2.276.3 related

The related specifies the second of the two Mated_item_association (see 4.2.275) objects related by the Mated_item_relationship.

See 4.3.780 for the application assertion.

4.2.276.4 relating

The relating specifies the first of the two Mated_item_association (see 4.2.275) objects related by the Mated_item_relationship.

See 4.3.780 for the application assertion.

4.2.276.5 transformation

The transformation specifies the Transformation (see 4.2.472) that shall be used to place the Geometric_model (see 4.2.211) representing the shape of the Item_instance (see 4.2.241) obtained by the related Mated_item_association relative to the Geometric_model representing the shape of the Item_instance obtained by the relating Mated_item_association.

In the case where the transformation is realized by a Template_instance (see 4.2.453) the scale factor of that Template_instance shall be omitted or set to 1.

See 4.3.779 and 4.3.783 for the application assertions.

The transformation need not be specified for a particular Mated_item_relationship.

If present, there shall be exactly one object that defines the transformation for a Mated_item_-relationship.

4.2.277 Material

A Material is the substance out of which an Item (see 4.2.236) is or can be made.

NOTE – For the specification of the raw material from which an Item is made the object Item is used with an attached Specific_classification (see 4.2.414) with classification type 'item role' and classification name 'raw material'. That object represents the material item with possibly additional information, like identification of supplier, quantity, size of the raw stock etc. and it can be used in corresponding relationships such as Make_from_relationship (see 4.2.273) or for a Process_operation (see 4.2.336).

The data associated with a Material are the following:

- described_object;
- material_property;
- name.

4.2.277.1 described_object

The described_object specifies the object a Material is applied to.

See 4.3.784, 4.3.785, 4.3.786, 4.3.787, 4.3.788, 4.3.789, 4.3.790, 4.3.791, 4.3.792, 4.3.793, 4.3.795, 4.3.796, 4.3.797, 4.3.798, 4.3.799, 4.3.800, 4.3.801, and 4.3.802 for the application assertions.

There shall be exactly one object that defines the described_object for a Material.

4.2.277.2 material_property

The material_property specifies a Material_property_representation (see 4.2.280) that describes the Material.

See 4.3.794 for the application assertion.

4.2.277.3 name

The name specifies the word or group of words by which the Material is referred to.

4.2.278 Material_condition

A Material_condition is the representation of a condition for a part surface wherein it contains the amount of material. The usage of material conditions is defined in ISO 2692.

NOTE – Tolerances of position are determined by a material condition of mating surfaces.

The data associated with a Material_condition are the following:

- condition.

The condition specifies the kind of Material_condition

EXAMPLE 146 – Maximum material condition (MMC) and least material condition (LMC) are examples for a Material_condition.

4.2.279 Material_property

A Material_property is a characteristic that depends on material aspects. The specification of the property is independent from the material definition for a part or tool.

A Material_property is a type of Item_property (see 4.2.243).

The data associated with a Material_property are the following:

- description;
- material_property_value;
- property_name.

4.2.279.1 description

The description specifies additional information about the Material_property.

EXAMPLE 147 – 'elastic modulus', 'yield point', 'wear rate', and 'corrosion behaviour' are examples for the description.

See 4.3.804 for the application assertion.

The description need not be specified for a particular Material_property.

If present, there shall be exactly one object that defines the description for a Material_property.

4.2.279.2 material_property_value

The material_property_value specifies the value for the Material_property.

See 4.3.803 for the application assertion.

4.2.279.3 property_name

The property_name specifies the kind of Material_property.

EXAMPLE 148 – 'EM1' is an example for a property_name that may be associated with a Material_property that is described as 'elastic modulus'.

4.2.280 Material_property_representation

A Material_property_representation is the representation of material related physical or technological properties and characteristics.

The data associated with a Material_property_representation are the following:

- environment_condition;
- property_value.

4.2.280.1 environment_condition

The environment_condition specifies the Data_environment (see 4.2.110) under which specific values are valid for particular properties.

See 4.3.805 for the application assertion.

4.2.280.2 property_value

The property_value specifies the Property_value (see 4.2.364) that represents the value for the Material_property_representation (see 4.2.280).

NOTE – The interpretation of this value, e.g., whether the value is required or calculated, is dependent on the semantics of the attribute 'value_interpretation' of the referenced Property_value.

See 4.3.806 for the application assertion.

4.2.281 Material_side_specification

A Material_side_specification is the information necessary to identify that side of a sheet like object where the material is.

The data associated with a Material_side_specification are the following:

- direction;
- is_defined_in;

– position.

4.2.281.1 direction

The direction specifies a vector that indicates the material direction.

4.2.281.2 is_defined_in

The is_defined_in specifies the Cartesian_coordinate_space (see 4.2.44) in which the position and the direction of the Material_side_specification are defined.

NOTE – Usually this is the same instance of Cartesian_coordinate_space as that of the object that has the corresponding Thickness (see 4.2.456) property assigned to it.

See 4.3.807 for the application assertion.

4.2.281.3 position

The position specifies a point on the surface of the sheet like object used as a start point to specify the material direction.

See 4.3.808 for the application assertion.

4.2.282 Mating_definition

A Mating_definition is a view of an Item_version (see 4.2.248) defining the physical connection of two or more Item_instance (see 4.2.241) objects including technical information about the kind of connection. This information is independent from the hierarchical assembly structure.

A Mating_definition is a type of Design_discipline_item_definition (see 4.2.134).

The data associated with a Mating_definition are the following:

– mating_type.

The mating_type specifies the kind of mating, i.e., how the items shall be mated together. Where applicable the following values shall be used:

– 'bolted joint': Attachment of two or more parts that are screwed together using screws, bolt nuts, and washers;

– 'brazing': Attachment of two or more parts established by soldering or brazing;

– 'clasping': Attachment of two or more parts established by using a cramp at one common outer border;

– 'clinching': Attachment of two or more parts established by pressing these parts together and stamping texture at the the point of contact;

– 'clipping': Attachment of two or more parts established by a clip (e.g. fixing the inner door panel at the inner door plate);

– 'doweling': Attachment of two or more parts using a pin that is fit into each of the parts by pressing;

– 'flanging': Attachment of two or more parts by crimping a projected length of one part over the edge of the other part;

– 'foam injection': Attachment of two or more parts where foam is injected into a cavity to establish the connection;

– 'glueing': Attachment of two parts established by adhesion material;

– 'laser-welding': Attachment of two or more parts that are welded together using laser technique;

– 'press fit': Attachment of two or more parts obtained by forces resulting from the selected size tolerances;

– 'riveting': Attachment of two or more parts that are riveted together at a certain spot using a rivet;

– 'sewing': Attachment of two or more parts that are sewn together (e.g. upholstering cloth for seat pillow);

– 'spot welding': Attachment of two or more parts that are welded together at a certain welding spot;

– 'stapling': Attachment of two or more parts at a certain spot using a nail or a wire stitch to establish the connection;

– 'welding': Attachment of two or more parts by welding together with one weld bead.

4.2.283 Model_change

A Model_change is a mechanism to describe the differences between two objects concerning the models describing these objects.

The data associated with a Model_change are the following:

– added_element;

– change_description;

– deleted_element;

– description;

– id;

– is_describing.

4.2.283.1 added_element

The added_element specifies the Detailed_element (see 4.2.135) that has been added to the relating model of the described model relationship.

See 4.3.810 for the application assertion.

4.2.283.2 change_description

The change_description specifies the model relationship that identifies the original (relating) model and the changed (related) model.

See 4.3.811, 4.3.813, and 4.3.814 for the application assertions.

There shall be exactly one object that defines the change_description for a Model_change.

4.2.283.3 deleted_element

The deleted_element specifies the Detailed_element (see 4.2.135) that has been deleted from the relating model of the described model relationship and therefore do not exist in the related model.

See 4.3.810 for the application assertion.

4.2.283.4 description

The description specifies additional information about the Model_change.

See 4.3.812 for the application assertion.

There shall be exactly one object that defines the description for a Model_change.

4.2.283.5 id

The id specifies the identifier of the Model_change.

The id need not be specified for a particular Model_change.

4.2.283.6 is_describing

The is_describing specifies the Change (see 4.2.54) that collects the Model_change objects and the Property_change (see 4.2.362) objects.

See 4.3.809 for the application assertion.

4.2.284 Model_image

A Model_image is a mechanism to generate a two-dimensional picture from two-dimensional or three-dimensional geometric or annotation elements. The mechanism is defined by clipping, scaling, and projecting information.

The data associated with a Model_image are the following:

- clipping;
- is_displaying;
- is_presented_in;
- name;
- scale;
- view_projection;
- view_volume_placement.

4.2.284.1 clipping

The clipping specifies the mathematical information necessary to define a two-dimensional area (clipping box) or a three-dimensional volume (clipping volume) that encloses all viewable geometric or annotation elements of a Styled_model (see 4.2.429) or of an External_model (see 4.2.166). Only those elements, or portions of elements, that fall within this two-dimensional area or three-dimensional volume shall be displayed.

The clipping need not be specified for a particular Model_image.

4.2.284.2 is_displaying

The is_displaying specifies the Styled_model (see 4.2.429) or the External_model (see 4.2.166) that contains the information to be processed for display.

See 4.3.815 and 4.3.817 for the application assertions.

There shall be exactly one object that the Model_image is_displaying.

4.2.284.3 is_presented_in

The is_presented_in specifies the View (see 4.2.490) within a Drawing_sheet (see 4.2.152) or View_area (see 4.2.491) the picture defined by the Model_image is presented in.

See 4.3.818 for the application assertion.

4.2.284.4 name

The name specifies the word or group of words by which the Model_image is referred to.

See 4.3.816 for the application assertion.

The name need not be specified for a particular Model_image.

If present, there shall be exactly one object that defines the name for a Model_image.

4.2.284.5 scale

The scale specifies the ratio between the size of the elements as defined in the shape model and the size of the elements as presented in the view. The default value for scale is 1.0.

4.2.284.6 view_projection

The view_projection specifies the type of projection that shall be either 'central' or 'parallel'.

4.2.284.7 view_volume_placement

The view_volume_placement specifies the transformation of a geometric or annotation element that is defined in the coordinate space of the Styled_model (see 4.2.429) to a target element in the coordinated space of the View (see 4.2.490) in which the Styled_model is presented.

4.2.285 Moments_of_inertia

A Moments_of_inertia describes the law of inertia in the context of the rotation of a rigid body. The moments of inertia I are described with a symmetric second level tensor:

$$I = \begin{pmatrix} I_xx & -I_xy & -I_xz \\ -I_yx & I_yy & -I_yz \\ -I_zx & -I_zy & I_zz \end{pmatrix}$$

A Moments_of_inertia is a type of Shape_dependent_property (see 4.2.396).

The data associated with a Moments_of_inertia are the following:

- inertia_value;
- is_defined_in;
- value_interpretation.

4.2.285.1 inertia_value

The inertia_value specifies 9 Value_with_unit (see 4.2.485) objects that are the values in a symmetric second level tensor.

See 4.3.820 for the application assertion.

4.2.285.2 is_defined_in

The is_defined_in specifies the Cartesian_coordinate_space (see 4.2.44) that the Moments_of_-inertia is defined in.

See 4.3.819 for the application assertion.

4.2.285.3 value_interpretation

The value_interpretation specifies information on how the Moments_of_inertia shall be interpreted.

NOTE – The Moments_of_inertia may be specified in the design stage of a product but it may also be documented as determined by measurements on prototypes.

EXAMPLE 149 – The value_interpretation may be e.g., 'required', 'designed', 'estimated', or 'measured'.

4.2.286 Multi_language_string

A Multi_language_string represents text information, stored in one or more languages, that is associated with objects.

The data associated with a Multi_language_string are the following:

– language_dependent_string.

The language_dependent_string specifies the String_with_language (see 4.2.425) that represents the text information in a particular language.

See 4.3.821 for the application assertion.

4.2.287 Next_higher_assembly

A Next_higher_assembly is a relationship where the attribute related specifies a constituent of an assembly and the attribute relating specifies the immediate parent assembly of the constituent. A constituent may be a single part or an assembly.

EXAMPLE 150 – The assembly carburettor may be assembled of single parts. The power train assembly is formed of assemblies.

A Next_higher_assembly is a type of Assembly_component_relationship (see 4.2.26).

4.2.288 Ngon_profile

A Ngon_profile is a profile that is specified by an enclosed area bounded by 3 or more connected straight lines. One side of the Ngon_profile is parallel to the x-axis of the local coordinate system of the profile respectively of the Axis_placement (see 4.2.33) serving as parameter_reference. This side intersects the y-axis at a negative value as shown in figure 41. The centre of the Ngon_profile is located in the origin.

A Ngon_profile is a type of Closed_profile (see 4.2.71).

The data associated with a Ngon_profile are the following:

- circumscribed_radius;
- corner_radius;
- number_of_sides.

4.2.288.1 circumscribed_radius

The circumscribed_radius specifies the radius of the circle circumscribing the Ngon_profile.

See 4.3.822 for the application assertion.

4.2.288.2 corner_radius

The corner_radius specifies the size of an arc blend between any two sides of the Ngon_profile. The corner_radius shall be smaller than $r \cdot cos(\frac{\pi}{n})$ where r is the circumscribed_radius and n is the number_of_sides.

See 4.3.822 for the application assertion.

The corner_radius need not be specified for a particular Ngon_profile.

4.2.288.3 number_of_sides

The number_of_sides specifies how many sides are needed for the Ngon_profile.

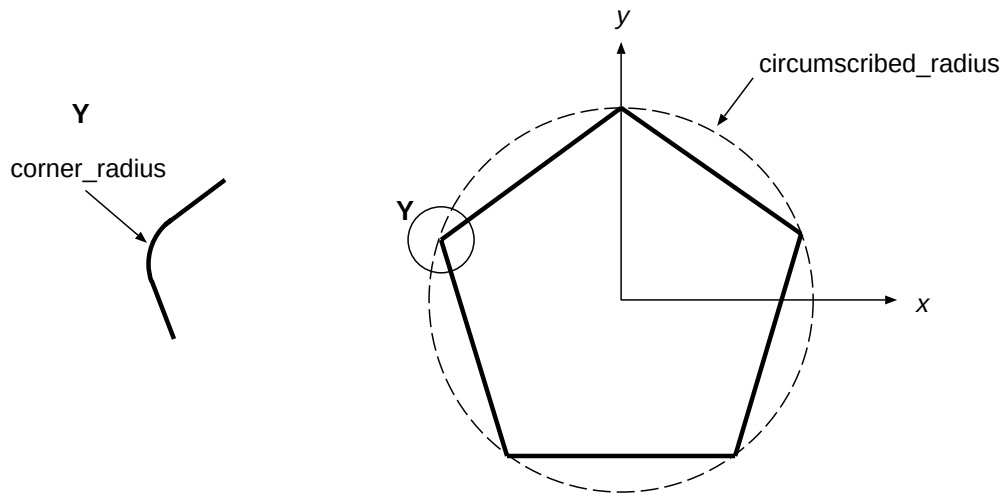


Figure 41 – Ngon_profile

4.2.289 Offset_curve

An Offset_curve is a curve in 2D or 3D space that is defined such that each point is at a fixed distance from the corresponding point on its basis curve. Furthermore the distance is perpendicular to the tangent vector at each point of the curve. The direction of the offset shall be defined. An Offset_curve shall not self-intersect and may reference any type of curve except another Offset_curve.

An Offset_curve is a type of Curve (see 4.2.102).

4.2.290 Offset_surface

An Offset_surface is a surface that is defined by offsetting a base surface by a distance along the surface normal of the basis surface.

An Offset_surface shall not self-intersect. The offset distance shall not exceed the radius of curvature in any direction at any point of the base surface, i.e., the surface shall not contain any ridge or singular point. The base surface shall not be a rectangular composite surface or another offset surface.

An Offset_surface is a type of Surface (see 4.2.434).

4.2.291 Open_bead_end_type

An Open_bead_end_type is an end condition of a bead that shall pass through the end of the part.

An Open_bead_end_type is a type of Bead_end_type (see 4.2.38).

4.2.292 Open_profile

An Open_profile is specified by an outline with two open ends.

An Open_profile is a type of Profile (see 4.2.355).

Each Open_profile is either a General_open_profile (see 4.2.202), a Tee_profile (see 4.2.452), a Vee_profile (see 4.2.487), a Square_u_profile (see 4.2.423), a Partial_circular_profile (see 4.2.303), or a Rounded_u_profile (see 4.2.388).

4.2.293 Open_slot_end_type

An Open_slot_end_type is an end condition of a slot specifying that the slot passes through the end of the shape of the part at this end.

An Open_slot_end_type is a type of Slot_end_type (see 4.2.406).

4.2.294 Organization

An Organization is a group of people involved in a particular business process.

The data associated with an Organization are the following:

- delivery_address;
- id;
- name;
- organization_type;
- postal_address;
- visitor_address.

4.2.294.1 delivery_address

The delivery_address specifies the address where to deliver goods to.

The delivery_address need not be specified for a particular Organization.

4.2.294.2 id

The id specifies the identifier of the Organization.

NOTE – The assignment of this attribute is usually controlled by a registration authority. The registration authority may be a public organization that assigns identifiers to corporations, or it may be the parent corporation that assigns component identifiers to its components.

EXAMPLE 151 – The Organization may be the code assigned to the Organization for a listing in a stock market, or it may be a department number.

4.2.294.3 name

The name specifies the word or group of words used to refer to the Organization.

4.2.294.4 organization_type

The organization_type specifies the type of the Organization.

EXAMPLE 152 – 'company', 'plant', or 'department' are examples for organization_type.

4.2.294.5 postal_address

The postal_address specifies the address for letter mail.

The postal_address need not be specified for a particular Organization.

4.2.294.6 visitor_address

The visitor_address specifies the address where the organization receives visitors.

The visitor_address need not be specified for a particular Organization.

4.2.295 Organization_in_contract

An Organization_in_contract is a mechanism to associate the person who is signing a contract and the organization which the person is signing for with a Contract (see 4.2.99).

The data associated with an Organization_in_contract are the following:

- contract;
- contracted_organization;
- role_of_organization;
- signature.

4.2.295.1 contract

The contract specifies the Contract (see 4.2.99) the Organization_in_contract participates in.

See 4.3.823 for the application assertion.

4.2.295.2 contracted_organization

The contracted_organization specifies the organization that participates in the contract.

See 4.3.826 for the application assertion.

4.2.295.3 role_of_organization

The role_of_organization specifies the function performed by the signing organization with respect to the contract.

EXAMPLE 153 – 'customer' or 'owner' are examples for role_of_organization.

See 4.3.825 for the application assertion.

There shall be exactly one object that defines the role_of_organization for an Organization_in_contract.

4.2.295.4 signature

The signature specifies the Date_and_person_or_organization (see 4.2.112) representing the person who signed the contract on behalf of the contracted organization and the date of the signature.

See 4.3.824 for the application assertion.

4.2.296 Organization_relationship

An Organization_relationship is a relationship between two organizations.

EXAMPLE 154 – A team belongs to a department which itself belongs to a company. This organizational structure may be built up using the Organization_relationship mechanism.

The data associated with an Organization_relationship are the following:

- related;
- relating;
- relation_type.

4.2.296.1 related

The related specifies the second Organization (see 4.2.294) in an Organization_relationship.

NOTE – The semantic of this attribute is defined by the attribute 'relation_type'.

See 4.3.827 for the application assertion.

4.2.296.2 relating

The relating specifies the one organization in an Organization_relationship.

NOTE – The semantic of this attribute is defined by the attribute 'relation_type'.

See 4.3.827 for the application assertion.

4.2.296.3 relation_type

The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:

- 'hierarchy': The related Organization (see 4.2.294) is a sub organization of the relating Organization;
- 'correspondance': The related Organization is actually the same as the relating Organization;
- 'succession': The related Organization is the successor of the relating Organization.

4.2.297 Oriented_edge_curve

An Oriented_edge_curve is a bounded segment of a curve. The Oriented_edge_curve is of finite length and is defined from the edge_start vertex to the edge_end vertex following the curve defined

by the edge_geometry. Additionally an Oriented_edge_curve has an associated orientation that specifies the mathematical sense of an Oriented_edge_curve. The Oriented_edge_curve shall not self-intersect.

The data associated with an Oriented_edge_curve are the following:

- edge_end;
- edge_geometry;
- edge_start.

4.2.297.1 edge_end

The edge_end specifies the second and last vertex of the edge. In the case of an edge that forms a close loop the edge_end will be identical to the edge_start.

See 4.3.836 for the application assertion.

4.2.297.2 edge_geometry

The edge_geometry specifies a curve that represents the shape of an edge. The extent of the edge is confined to that section of the curve between the start and end vertex.

See 4.3.828, 4.3.829, 4.3.830, 4.3.831, 4.3.832, 4.3.833, 4.3.834, and 4.3.835 for the application assertions.

There shall be exactly one object that defines the edge_geometry for an Oriented_edge_curve.

4.2.297.3 edge_start

The edge_start specifies the first vertex of an edge.

See 4.3.836 for the application assertion.

4.2.298 Pair_value

A Pair_value is one particular value of the pair parameter of a Kinematic_pair (see 4.2.258). A Pair_value shall represent either a cylindrical pair value, a gear pair value, a planar pair value, a prismatic pair value, a rack and pinion pair value, a revolute pair value, a rolling curve pair value, a rolling surface pair value, a screw pair value, a sliding curve pair value, a sliding surface pair value, a spherical pair value, or an universal pair value.

The data associated with a Pair_value are the following:

- is_applied_to.

The is_applied_to specifies the Kinematic_pair (see 4.2.258) that the Pair_value applies to.

See 4.3.837 for the application assertion.

4.2.299 Panel_hole_with_barring

A Panel_hole_with_barring is a hole in a panel where the material is not completely punched out but partially bended in the punch direction.

EXAMPLE 155 – Figure 42 shows an example of a Panel_hole_with_barring.

A Panel_hole_with_barring is a type of Hole_feature_in_panel (see 4.2.229).

The data associated with a Panel_hole_with_barring are the following:

– barring_profile.

The barring_profile specifies the Closed_profile (see 4.2.71) that represents that portion of the panel that is actually punched out.

NOTE – The difference in area between the barring_profile and the hole_profile is that portion of the panel that is bended at 90 degrees in punch direction.

See 4.3.838 for the application assertion.

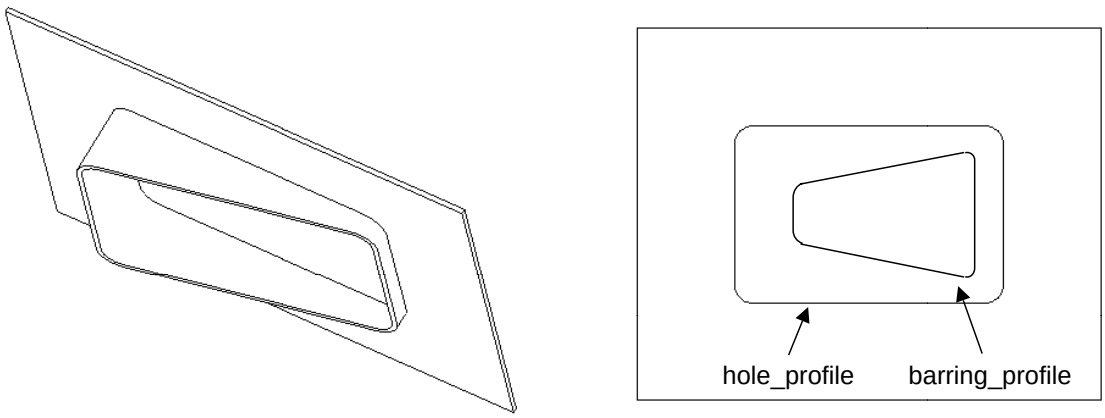


Figure 42 – Panel_hole_with_barring

4.2.300 Parabola

A Parabola is an open conic curve defined by its focal length (distance between focal point and apex), apex and the direction of the line joining the apex and focus.

A Parabola is a type of Curve (see 4.2.102).

4.2.301 Parallelism_tolerance

A Parallelism_tolerance is a tolerance for the condition of a surface equidistant at all points from a datum plane or of an axis equidistant along its length to a datum axis. A Parallelism_tolerance specifies one of the following:

– a tolerance zone by two planes or lines (if the attribute 'affected_plane' is specified) parallel to a datum plane, or axis, within which the line elements of the surface or axis of the considered feature must lie;

– a cylindrical tolerance zone whose axis is parallel to a datum axis within which the axis of the considered feature must lie.

A Parallelism_tolerance is a type of Geometric_tolerance (see 4.2.216).

The data associated with a Parallelism_tolerance are the following:

– affected_plane;

– datum;

– segment_size.

4.2.301.1 affected_plane

The affected_plane specifies the plane a certain type of geometric tolerance applies in. This attribute covers information that is given on drawings implicitly by the drawing plane. If the affected_plane attribute is not specified the tolerance applies in all directions which results in a cylindrical tolerance zone (on drawings indicated by the diameter symbol).

See 4.3.840 for the application assertion.

The affected_plane need not be specified for a particular Parallelism_tolerance.

4.2.301.2 datum

The datum specifies the Datum_feature (see 4.2.117) or the Compound_datum (see 4.2.80) that defines the reference for this type of geometric tolerance.

See 4.3.839 for the application assertion.

4.2.301.3 segment_size

The segment_size specifies the length (or the edge length) of a segment if the geometric tolerance is not applied to the total length/area of an axis/surface but on a unit base (e.g. to each segment with a length of 20 mm).

The segment_size need not be specified for a particular Parallelism_tolerance.

4.2.302 Partial_circular_path

A Partial_circular_path is a path that is specified by a direction of travel along an arc of constant radius around an axis. The Partial_circular_path begins on the x-axis of the Axis_placement (see 4.2.33) and travels in positive direction around the z-axis.

EXAMPLE 156 – Figure 43 illustrates a Partial_circular_path.

A Partial_circular_path is a type of Circular_path (see 4.2.60).

The data associated with a Partial_circular_path are the following:

- sweep_angle.

The sweep_angle specifies the angle that defines the arc shaped path.

See 4.3.841 for the application assertion.

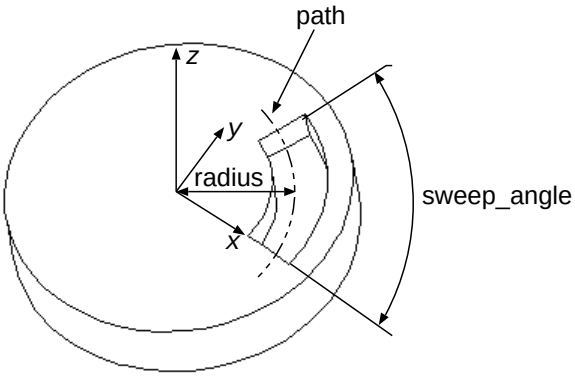


Figure 43 – Partial_circular_path

4.2.303 Partial_circular_profile

A Partial_circular_profile is a profile that is specified by a circular arc. The Partial_circular_profile starts at the reference x-axis and travels around the reference z-axis in positive direction.

EXAMPLE 157 – Figure 44 illustrates a Partial_circular_profile.

A Partial_circular_profile is a type of Open_profile (see 4.2.292).

The data associated with a Partial_circular_profile are the following:

- radius;
- sweep_angle.

4.2.303.1 radius

The radius specifies the radius of the arc to define a Partial_circular_profile.

See 4.3.842 for the application assertion.

4.2.303.2 sweep_angle

The sweep_angle specifies the size of the angle to define a Partial_circular_profile.

See 4.3.842 for the application assertion.

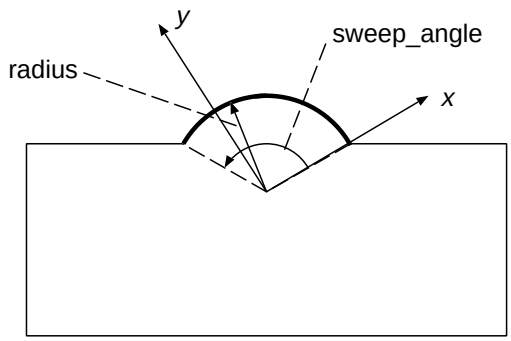


Figure 44 – Partial_circular_profile

4.2.304 Path

A Path is the representation of a continuous direction of travel.

Each Path is either a General_path (see 4.2.203), a Linear_path (see 4.2.268), or a Circular_path (see 4.2.60).

The data associated with a Path are the following:

– is_defined_in.

The is_defined_in specifies the Cartesian_coordinate_space (see 4.2.44) the Path is defined in. In the case where the Path is used to define an Included_feature (see 4.2.232) this coordinate space shall be the same as the part's coordinate space. In the case where the Path is used to define a feature that is applied as a Placed_feature (see 4.2.313) this coordinate space shall be the same as the feature's coordinate space.

See 4.3.843 for the application assertion.

4.2.305 Path_approximation

A Path_approximation is a motion parameter measure with or without units by which different discrete poses in the course of a kinematics motion are distinguished.

NOTE – In order to introduce time aspects the motion parameter may be defined with time units.

4.2.306 Perpendicularity_tolerance

A Perpendicularity_tolerance is a tolerance for the condition of a surface, median plane, or axis at a right angle to a datum plane or axis. A Perpendicularity_tolerance specifies one of the following:

– a tolerance zone defined by two parallel planes perpendicular to a datum plane, or axis, within which the surface or median plane of the considered feature must lie;

– a tolerance zone defined by two parallel planes perpendicular to a datum axis within which the axis of the considered feature must lie;

– a cylindrical tolerance zone (if the attribute 'affected_plane' is not specified) perpendicular to a datum plane within which the axis of the considered feature must lie;

– a tolerance zone defined by two parallel lines perpendicular to a datum plane, or axis, within which an element of the surface must lie.

A Perpendicularity_tolerance is a type of Geometric_tolerance (see 4.2.216).

The data associated with a Perpendicularity_tolerance are the following:

– affected_plane;

– datum;

– segment_size.

4.2.306.1 affected_plane

The affected_plane specifies the plane a certain type of geometric tolerance applies in. This attribute covers information that is given on drawings implicitly by the drawing plane. If the affected_plane attribute is not specified the tolerance applies in all directions which results in a cylindrical tolerance zone (on drawings indicated by the diameter symbol).

See 4.3.845 for the application assertion.

The affected_plane need not be specified for a particular Perpendicularity_tolerance.

4.2.306.2 datum

The datum specifies the Datum_feature (see 4.2.117) or the Compound_datum (see 4.2.80) that defines the reference for this type of geometric tolerance.

See 4.3.844 for the application assertion.

4.2.306.3 segment_size

The segment_size specifies the length (or the edge length) of a segment if the geometric tolerance is not applied to the total length/area of an axis/surface but on a unit base (e.g. to each segment with a length of 20 mm).

The segment_size need not be specified for a particular Perpendicularity_tolerance.

4.2.307 Person

A Person is an individual human being who has some relationship to the product data. The Person shall always be identified in the context of one or more organizations.

The data associated with a Person are the following:

– address;

- id;
- name.

4.2.307.1 address

The address specifies the location of the office of the Person.

The address need not be specified for a particular Person.

4.2.307.2 id

The id specifies an identifier of the Person. The identifier shall be unique within the scope of an organization.

EXAMPLE 158 – The id may be a staff number or a user id in a computer system.

4.2.307.3 name

The name specifies the word or group of words used to refer to the Person.

NOTE – The name includes the first, middle, and last names as well as titles, if applicable.

The name need not be specified for a particular Person.

4.2.308 Person_in_organization

A Person_in_organization is the specification of a Person (see 4.2.307) in the context of an Organization (see 4.2.294).

The data associated with a Person_in_organization are the following:

- organization;
- person;
- role.

4.2.308.1 organization

The organization specifies the Organization (see 4.2.294) with which the Person (see 4.2.307) is associated.

See 4.3.846 for the application assertion.

4.2.308.2 person

The person specifies the Person (see 4.2.307).

See 4.3.847 for the application assertion.

4.2.308.3 role

The role specifies the relationship between the Person (see 4.2.307) and the Organization (see 4.2.294).

4.2.309 Person_or_organization_assignment

A Person_or_organization_assignment is an object that associateds an Organization (see 4.2.294) a Person (see 4.2.307), or a Person_in_organization (see 4.2.308) with an object.

The data associated with a Person_or_organization_assignment are the following:

- assigned_person_or_organization;
- is_applied_to;
- role.

4.2.309.1 assigned_person_or_organization

The assigned_person_or_organization specifies the concerned Organization (see 4.2.294), the Person (see 4.2.307), or the Person_in_organization (see 4.2.308).

See 4.3.870, 4.3.871, and 4.3.872 for the application assertions.

There shall be exactly one object that defines the assigned_person_or_organization for a Person_or_organization_assignment.

4.2.309.2 is_applied_to

The is_applied_to specifies the object with which the Person_or_organization_assignment is associated.

See 4.3.848, 4.3.849, 4.3.850, 4.3.851, 4.3.852, 4.3.853, 4.3.854, 4.3.855, 4.3.856, 4.3.857, 4.3.858, 4.3.859, 4.3.860, 4.3.861, 4.3.862, 4.3.863, 4.3.864, 4.3.865, 4.3.866, 4.3.867, 4.3.868, 4.3.869, 4.3.873, 4.3.874, 4.3.875, 4.3.876, 4.3.877, 4.3.878, 4.3.879, 4.3.880, 4.3.881, 4.3.882, 4.3.883, 4.3.884, 4.3.885, 4.3.886, 4.3.887, 4.3.888, and 4.3.889 for the application assertions.

There shall be exactly one object that the Person_or_organization_assignment is_applied_to.

4.2.309.3 role

The role specifies the responsibility of the assigned Person (see 4.2.307) or Organization (see 4.2.294) with respect to the object that it is applied to. Where applicable the following values shall be used:

- 'creator': The referenced object has been created by the Person or Organization;
- 'editor': The assigned Person or Organization is responsible for making any changes to any attribute of the referenced object;

– 'owner': The assigned Person or Organization owns the referenced object;

– 'customer': The assigned Person or Organization acts as a purchaser or consumer of the referenced object;

– 'custodian': The assigned Person or Organization is responsible for the existence and integrity of the referenced object.

4.2.310 Physical_instance

A Physical_instance is the denomination of a physically realized object that is identified either by a lot id or by a serial number.

The data associated with a Physical_instance are the following:

– description;

– is_based_on;

– lot_id;

– serial_number.

4.2.310.1 description

The description specifies additional information about the Physical_instance.

See 4.3.891 for the application assertion.

The description need not be specified for a particular Physical_instance.

If present, there shall be exactly one object that defines the description for a Physical_instance.

4.2.310.2 is_based_on

The is_based_on specifies the Product_specification (see 4.2.354) or the Item_version (see 4.2.248) that collects the information defining the Physical_instance.

See 4.3.890 and 4.3.892 for the application assertions.

There shall be exactly one object that the Physical_instance is_based_on.

4.2.310.3 lot_id

The lot_id specifies the identifier of the lot the Physical_instance is part of.

The lot_id need not be specified for a particular Physical_instance.

4.2.310.4 serial_number

The serial_number is an identifier that distinguishes one Physical_instance from another.

The serial_number need not be specified for a particular Physical_instance.

4.2.311 Physical_instance_usage

A Physical_instance_usage is a mechanism to relate one Physical_instance (see 4.2.310) as a component to another Physical_instance that plays the role of an assembly.

The data associated with a Physical_instance_usage are the following:

- physical_assembly;
- physical_component;
- role.

4.2.311.1 physical_assembly

The physical_assembly specifies the Physical_instance (see 4.2.310) that serves as the assembly in the physical structure.

See 4.3.894 for the application assertion.

4.2.311.2 physical_component

The physical_component specifies the Physical_instance (see 4.2.310) that serves as a component in the physical structure.

See 4.3.894 for the application assertion.

4.2.311.3 role

The role specifies the Item_instance (see 4.2.241) the physical component is the realization of.

See 4.3.893 for the application assertion.

4.2.312 Physical_model

A Physical_model is a model of the shape of a part made from, e.g., clay or wood in a specific scale.

> EXAMPLE 159 – A Physical_model may be used to get an impression of the future outlook of, e.g., a car or to use it, e.g., in a wind tunnel. Furthermore physical models are needed for cores of cast iron parts or for collision detection between parts or tools.

A Physical_model is a type of External_document (see 4.2.162).

The data associated with a Physical_model are the following:

- found_in;
- scale.

4.2.312.1 found_in

The found_in specifies information about the location where the Physical_model is kept.

EXAMPLE 160 – The found_in may be an address of an organization, e.g., of the storage where the Physical_model is stored, or an address of the person responsible for the Physical_model.

4.2.312.2 scale

The scale specifies the scale of the physical model.

EXAMPLE 161 – '2' which equals a ratio of 2:1 and '0,2' which equals a ratio of 1:5 are examples for scale.

4.2.313 Placed_feature

A Placed_feature is the identification of a feature that is defined in its own coordinate space and placed into a reference coordinate space by a Feature_placement (see 4.2.178) that carries a geometric Transformation (see 4.2.472).

NOTE 1 – A Placed_feature may or may not have shape representation.

NOTE 2 – A Placed_feature is supposed to have all information necessary to create the part's shape resulting from the application of the feature.

A Placed_feature is a type of Feature_occurrence (see 4.2.176).

The data associated with a Placed_feature are the following:

- definition;
- is_placed_by.

4.2.313.1 definition

The definition specifies the Feature_definition (see 4.2.175) that has all information (type and parameters) that is needed to modify the part's shape as a result from the application of the feature.

See 4.3.895 for the application assertion.

4.2.313.2 is_placed_by

The is_placed_by specifies the Feature_placement (see 4.2.178) that is used to place the feature in a reference coordinate space.

See 4.3.896 for the application assertion.

4.2.314 Placed_target

A Placed_target is a target that is described implicitly by parameters with reference to its parameter_reference.

A Placed_target is a type of Datum_target (see 4.2.118).

Each Placed_target is either a Target_circle (see 4.2.447), a Target_point (see 4.2.449), a Target_rectangle (see 4.2.450), or a Target_line (see 4.2.448).

The data associated with a Placed_target are the following:

- is_defined_in;
- parameter_reference.

4.2.314.1 is_defined_in

The is_defined_in specifies the Cartesian_coordinate_space (see 4.2.44) that the Axis_placement (see 4.2.33) that serves as parameter_reference is defined in.

NOTE – A Placed_target is not defined in its own local and independent coordinate system.

See 4.3.898 for the application assertion.

4.2.314.2 parameter_reference

The parameter_reference specifies the Axis_placement (see 4.2.33) that the parameters or the Placed_target refer to.

See 4.3.897 for the application assertion.

4.2.315 Planar_bottom_condition

A Planar_bottom_condition is a bottom of a Pocket_feature_in_solid (see 4.2.321) that is flat.

A Planar_bottom_condition is a type of Pocket_bottom_condition (see 4.2.320).

The data associated with a Planar_bottom_condition are the following:

- floor_location;
- floor_normal;
- floor_radius.

4.2.315.1 floor_location

The floor_location specifies a Cartesian point that is defined in the same Cartesian_coordinate_space (see 4.2.44) as the concerned Pocket_feature_in_solid (see 4.2.321) and that shall be a point on the planar bottom.

4.2.315.2 floor_normal

The floor_normal specifies a direction vector that is defined in the same coordinate space as the concerned Pocket_feature_in_solid (see 4.2.321) and that indicates a direction normal to the planar bottom and pointing away from material.

4.2.315.3 floor_radius

The floor_radius specifies the blend radius at the bottom and the side edges of the Pocket_-feature_in_solid (see 4.2.321).

See 4.3.899 for the application assertion.

The floor_radius need not be specified for a particular Planar_bottom_condition.

4.2.316 Planar_top_condition

A Planar_top_condition is a top condition of a boss that is flat.

A Planar_top_condition is a type of Boss_top_condition (see 4.2.43).

The data associated with a Planar_top_condition are the following:

- top_location;
- top_normal.

4.2.316.1 top_location

The top_location specifies a Cartesian point that is defined in the same Cartesian_coordinate_-space (see 4.2.44) as the concerned Boss_feature (see 4.2.42) and that shall be a point on the planar top.

4.2.316.2 top_normal

The top_normal specifies a direction vector that is defined in the same coordinate space as the concerned Boss_feature (see 4.2.42) and that indicates a direction normal to the planar top and pointing away from material.

4.2.317 Plane

A Plane is an unbounded surface with a constant normal.

NOTE – A Plane is trimmed to a finite extension by applying a Face (see 4.2.167).

A Plane is a type of Surface (see 4.2.434).

4.2.318 Plus_minus_tolerance

A Plus_minus_tolerance is the specification of the allowable deviation from a theoretically exact geometric dimension. The Plus_minus_tolerance is represented by a delta value that is to be subtracted from the associated nominal value and by another delta value that is to be added to the associated nominal value.

The data associated with a Plus_minus_tolerance are the following:

- lower_bound;
- upper_bound.

4.2.318.1 lower_bound

The lower_bound specifies the value of the tolerance that shall be subtracted from the dimension value or from the feature parameter value to establish the minimum allowable deviation boundary.

4.2.318.2 upper_bound

The upper_bound specifies the value of the tolerance that shall be added to the dimension value or to the feature parameter value to establish the maximum allowable deviation boundary.

4.2.319 Plus_minus_toleranced_datum

A Plus_minus_toleranced_datum is an entry in a general tolerance table cell that is a plus minus tolerance range specified by a value with unit.

The data associated with a Plus_minus_toleranced_datum are the following:

- tolerance_value.

The tolerance_value specifies the value with unit that gives the tolerance range between the minus value and the plus value.

4.2.320 Pocket_bottom_condition

A Pocket_bottom_condition is the definition of a kind of bottom that characterizes a Pocket_feature_in_solid (see 4.2.321).

Each Pocket_bottom_condition is either a Through_bottom_condition (see 4.2.459), a Planar_bottom_condition (see 4.2.315), or a General_bottom_condition (see 4.2.191).

The data associated with a Pocket_bottom_condition are the following:

- is_applied_to.

The is_applied_to specifies to which end of the path the Pocket_bottom_condition applies. The following values shall be used:

– 'start': The Pocket_bottom_condition is applied to that end of the Pocket_feature_in_solid (see 4.2.321) that corresponds to the start of the defining Linear_path (see 4.2.268);

– 'end': The Pocket_bottom_condition is applied to that end of the Pocket_feature_in_solid that corresponds to the end of the defining Linear_path.

4.2.321 Pocket_feature_in_solid

A Pocket_feature_in_solid is a volume of defined shape to be removed from the part.

A Pocket_feature_in_solid is a type of Process_feature_in_solid (see 4.2.335).

Each Pocket_feature_in_solid is either a Rectangular_open_pocket_in_solid (see 4.2.373), a Rectangular_closed_pocket_in_solid (see 4.2.369), or a General_pocket_in_solid (see 4.2.205).

The data associated with a Pocket_feature_in_solid are the following:

– bottom_condition;

– profile_placement;

– sweep_path;

– taper.

4.2.321.1 bottom_condition

The bottom_condition specifies the Pocket_bottom_condition (see 4.2.320) that defines the type of the pocket bottom.

See 4.3.904 for the application assertion.

4.2.321.2 profile_placement

The profile_placement specifies a geometric Transformation (see 4.2.472) that may be used to place the defining Closed_profile (see 4.2.71) in the Pocket_feature_in_solid's coordinate space.

See 4.3.905 for the application assertion.

The profile_placement need not be specified for a particular Pocket_feature_in_solid.

4.2.321.3 sweep_path

The sweep_path specifies the Linear_path (see 4.2.268) that is used to sweep the Closed_profile (see 4.2.71).

See 4.3.903 for the application assertion.

4.2.321.4 taper

The taper specifies a variation of the size of the Pocket_feature_in_solid. The variation shall be specified either by an Angle_taper (see 4.2.10) or by a Directed_taper (see 4.2.145) but not by a Diameter_taper (see 4.2.138).

See 4.3.900, 4.3.901, and 4.3.902 for the application assertions.

The taper need not be specified for a particular Pocket_feature_in_solid.

If present, there shall be exactly one object that defines the taper for a Pocket_feature_in_solid.

4.2.322 Point

A Point is a location in a real Cartesian coordinate space.

A Point is a type of Detailed_geometric_model_element (see 4.2.136).

Each Point is either a Point_on_surface (see 4.2.328), a Point_on_curve (see 4.2.327), or a Cartesian_point (see 4.2.47).

4.2.323 Point_appearance

A Point_appearance governs the visual appearance of a point.

The data associated with a Point_appearance are the following:

- colour;
- marker.

4.2.323.1 colour

The colour specifies the colour of a point. The colour is either userdefined or predefined. Figure 19 shows the predefined colours.

4.2.323.2 marker

The marker specifies the point marker that shall be used to present a point. Figure 45 shows the predefined point markers.

4.2.324 Point_direction

A Point_direction is the location in space of a single axis.

EXAMPLE 162 – The Point_direction may be used to describe the position and orientation of a machining tool.

A Point_direction is a type of Detailed_geometric_model_element (see 4.2.136).

The data associated with a Point_direction are the following:

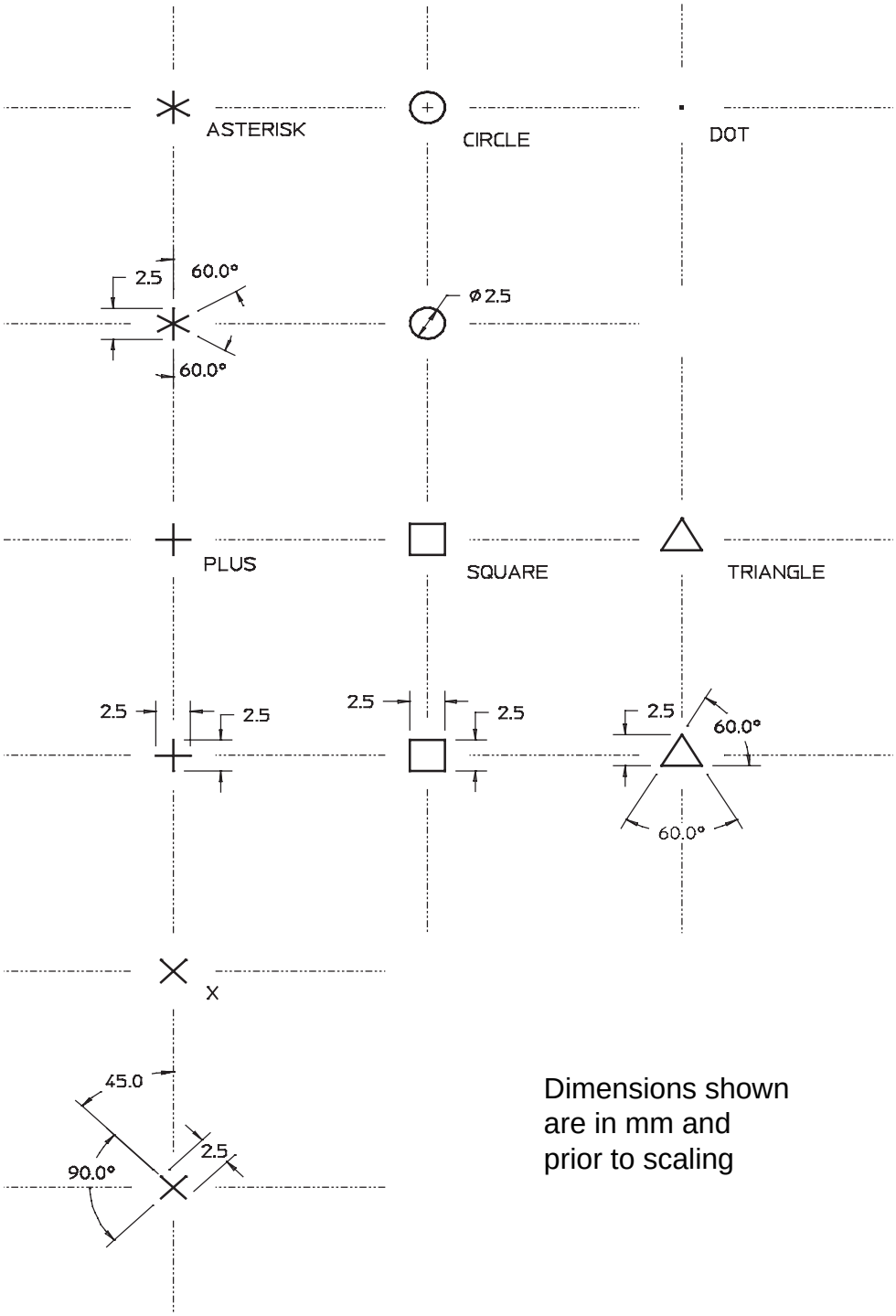


Figure 45 – Predefined point markers

- location;
- vector.

4.2.324.1 location

The location specifies the position of the Point_direction (see 4.2.324) in space.

See 4.3.906 for the application assertion.

4.2.324.2 vector

The vector specifies the direction of the Point_direction in space.

4.2.325 Point_direction_model

A Point_direction_model is a model in three dimensional space consisting only of ordered sequences of points or points with associated directions.

EXAMPLE 163 – For scanned data along planar intersections of a part, several point direction sequences can be used to specify the measured points together with the measured normal direction on the object's surface in their sequence along the planar intersection. Another example is the specification of a tool path where the sequence specifies the tool path as a polygon and the direction specifies the direction of the tool, e.g., a miller, in each point of the path.

A Point_direction_model is a type of Geometric_model (see 4.2.211).

The data associated with a Point_direction_model are the following:

- point;
- point_direction.

4.2.325.1 point

The point specifies the Point_sequence (see 4.2.329) elements of a Point_direction_model.

See 4.3.908 for the application assertion.

4.2.325.2 point_direction

The point_direction specifies the Point_direction_sequence (see 4.2.326) elements of a Point_direction_model.

See 4.3.907 for the application assertion.

4.2.326 Point_direction_sequence

A Point_direction_sequence is an ordered list of points and associated directions.

EXAMPLE 164 – A Point_direction_sequence may be used to represent the consecutive locations of a cutting tool during a machining operation.

A Point_direction_sequence is a type of Detailed_geometric_model_element (see 4.2.136).

The data associated with a Point_direction_sequence are the following:

– element.

The element specifies the Point_direction (see 4.2.324) objects or the Point_on_surface (see 4.2.328) objects that define the consecutive points and the appropriate axis directions. In the case of a Point_on_surface the direction is the cross product of the principal tangent vectors u and v.

See 4.3.909 and 4.3.910 for the application assertions.

There shall be at least one object that defines the element for a Point_direction_sequence.

4.2.327 Point_on_curve

A Point_on_curve is a point that is defined by evaluating a given curve at a specified parameter value.

A Point_on_curve is a type of Point (see 4.2.322).

4.2.328 Point_on_surface

A Point_on_surface is a point that is defined in the domain of a surface. Its location in 3D geometric space is found by evaluating the surface at a particular pair of parameter values.

NOTE – The object implicitly represents the topological relation between a point and a surface.

A Point_on_surface is a type of Point (see 4.2.322).

4.2.329 Point_sequence

A Point_sequence is an ordered list of points.

A Point_sequence is a type of Detailed_geometric_model_element (see 4.2.136).

The data associated with a Point_sequence are the following:

– element.

The element specifies the point elements that define the consecutive points.

See 4.3.911 for the application assertion.

4.2.330 Poly_loop

A Poly_loop is a loop with straight edges. It is represented by an ordered coplanar collection of Cartesian points forming the vertices of the Poly_loop. The loop is composed of straight line segments joining a point in the collection to the succeeding point in the collection. The closing

segment is from the last to the first point in the collection. The direction of the Poly_loop is in the direction of the line segments.

The data associated with a Poly_loop are the following:

- outer;
- polygon.

4.2.330.1 outer

The outer specifies whether the Poly_loop represents the outer boundary of the Faceted_face (see 4.2.174) or not.

4.2.330.2 polygon

The polygon specifies the points that are joined by line segments to form a Poly_loop (see 4.2.330).

See 4.3.912 for the application assertion.

4.2.331 Polyline

A Polyline is a curve that is represented by n-1 linear segments, defined by a list of n points, whereby two following points may be identical. A Polyline may be closed and may self-intersect.

A Polyline is a type of Curve (see 4.2.102).

4.2.332 Position_tolerance

A Position_tolerance is the definition of a zone within which the centre, axis, or centre plane of a feature of size is permitted to vary from true (theoretically exact) position. Basic dimensions establish the true position from specified datum features and between interrelated features. The concept of position tolerances is defined in ISO 5458.

A Position_tolerance is a type of Geometric_tolerance (see 4.2.216).

The data associated with a Position_tolerance are the following:

- affected_plane;
- datum.

4.2.332.1 affected_plane

The affected_plane specifies the plane a certain type of geometric tolerance applies in. This attribute covers information that is given on drawings implicitly by the drawing plane. If the affected_plane attribute is not specified the tolerance applies in all directions which results in a cylindrical tolerance zone (on drawings indicated by the diameter symbol).

EXAMPLE 165 – Figure 46 illustrates the meaning of the affected_plane in the case of a Position_-tolerance.

See 4.3.914 for the application assertion.

The affected_plane need not be specified for a particular Position_tolerance.

4.2.332.2 datum

The datum specifies the Datum_feature (see 4.2.117) or the Compound_datum (see 4.2.80) that defines the reference for this type of geometric tolerance.

See 4.3.913 for the application assertion.

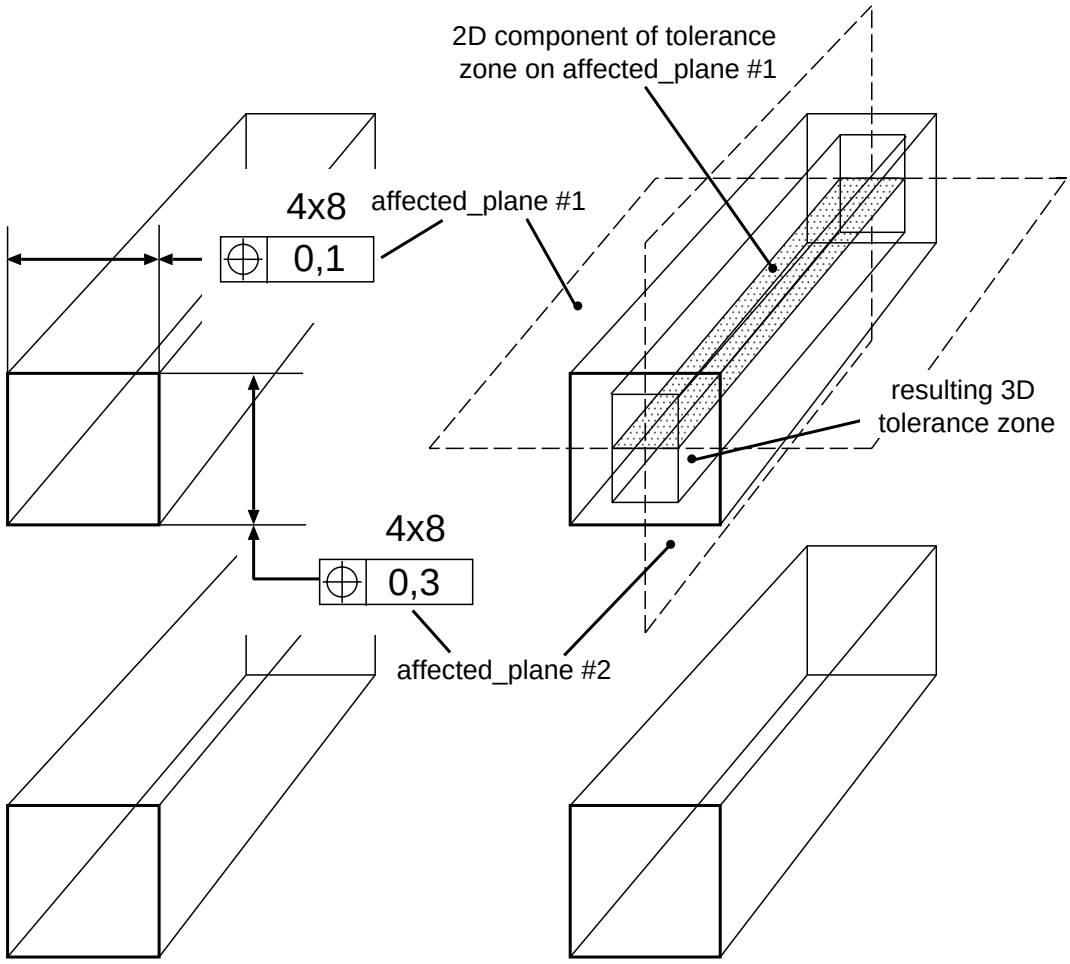


Figure 46 – Position tolerancing with affected planes

4.2.333 Process_cost_property

A Process_cost_property is a property of process related object that specifies the costs that are connected with it.

A Process_cost_property is a type of Process_property (see 4.2.345).

The data associated with a Process_cost_property are the following:

- cost_property_value.

The cost_property_value specifies the value of the Process_cost_property.

See 4.3.915 for the application assertion.

4.2.334 Process_feature_in_panel

A Process_feature_in_panel is a feature that is defined to be applied to a sheet like part.

A Process_feature_in_panel is a type of Feature_definition (see 4.2.175).

Each Process_feature_in_panel is either a Locator_feature (see 4.2.271), a Form_feature_in_panel (see 4.2.188), a Bead_feature (see 4.2.39), a Hole_feature_in_panel (see 4.2.229), a Compound_feature_in_panel (see 4.2.81), or a Joggle_feature (see 4.2.250).

The data associated with a Process_feature_in_panel are the following:

- punch_direction.

The punch_direction specifies the stamping direction when manufacturing the Process_feature_in_panel.

NOTE – Usually the punch_direction is the direction normal to the panel surface towards that side of the panel that shows the resulting Process_feature_in_panel.

The punch_direction need not be specified for a particular Process_feature_in_panel.

4.2.335 Process_feature_in_solid

A Process_feature_in_solid is a feature that is defined to be applied to a solid part. Each single Process_feature_in_solid (i.e. not a Compound_feature_in_solid (see 4.2.84)) is specified as a volume defined by a Profile (see 4.2.355) and a sweep Path (see 4.2.304); depending on the type of Process_feature_in_solid this volume is either added to or subtracted from the part.

In the case where the Process_feature_in_solid is defined to be applied as an Included_feature (see 4.2.232) the feature itself as well as the Path and the Profile shall be defined in the part's coordinate space. In the case where the Process_feature_in_solid is defined to be applied as a Placed_feature (see 4.2.313) the Process_feature_in_solid as well as the Profile shall be defined in their own local coordinate spaces; the Path shall be defined in the feature coordinate space. The spatial constraints between profile, feature, and part shall be specified by geometric Transformation (see 4.2.472) objects.

NOTE – The latter allows to re-use feature definitions for multiple occurrences of the feature.

A Process_feature_in_solid is a type of Feature_definition (see 4.2.175).

Each Process_feature_in_solid is either a Pocket_feature_in_solid (see 4.2.321), a Slot_feature (see 4.2.407), a Rib_feature (see 4.2.382), a Round_hole (see 4.2.387), a Boss_feature (see 4.2.42), or a Compound_feature_in_solid (see 4.2.84).

4.2.336 Process_operation

A Process_operation is a specific step in the manufacturing process of an Item_version (see 4.2.248). There may be descriptions associated to a Process_operation.

EXAMPLE 166 – In the case of a welding operation, the welding points as a part of a wireframe model can be associated to a process operation to describe the locations where the robot has to weld for a welding process operation.

The data associated with a Process_operation are the following:

- description;
- id;
- is_defined_in;
- location;
- name;
- process_duration;
- process_type.

4.2.336.1 description

The description specifies additional information about the Process_operation.

See 4.3.917 for the application assertion.

The description need not be specified for a particular Process_operation.

If present, there shall be exactly one object that defines the description for a Process_operation.

4.2.336.2 id

The id specifies the identifier of the Process_operation that shall be unique within the scope of the associated Process_plan_version (see 4.2.344).

4.2.336.3 is_defined_in

The is_defined_in specifies the Cartesian_coordinate_space (see 4.2.44) of the Process_operation in case none of the tools associated by Process_operation_input_or_output (see 4.2.337) plays the role of a reference tool that has the reference coordinate space.

See 4.3.916 for the application assertion.

The is_defined_in need not be specified for a particular Process_operation.

4.2.336.4 location

The location specifies the manufacturing location this Process_operation is planned for.

EXAMPLE 167 – A Process_operation customized for a specific machine combination, e.g., in a production site in Paris may not be applicable to the production site in Frankfurt.

EXAMPLE 168 – In case of a part needs a special kind of heat treatment, the process operation 'heat treatment' of a particular process plan may be done at a supplier's location, where the appropriate equipment is available, and the other process operations of this process plan are done at the manufacturer's location.

See 4.3.918 for the application assertion.

The location need not be specified for a particular Process_operation.

4.2.336.5 name

The name specifies the word or group of words by which the Process_operation is referred to.

EXAMPLE 169 – 'operation Z10.114' is an example for name.

See 4.3.917 for the application assertion.

The name need not be specified for a particular Process_operation.

If present, there shall be exactly one object that defines the name for a Process_operation.

4.2.336.6 process_duration

The process_duration specifies the time which elapses between the start of the Process_operation and the completion of the Process_operation.

See 4.3.919 for the application assertion.

The process_duration need not be specified for a particular Process_operation.

4.2.336.7 process_type

The process_type specifies the type of the Process_operation.

EXAMPLE 170 – 'stamping', 'milling', 'drilling', 'welding', 'casting', and 'painting' are examples for process_type.

4.2.337 Process_operation_input_or_output

A Process_operation_input_or_output is the input or output respectively of a Process_operation (see 4.2.336).

The data associated with a Process_operation_input_or_output are the following:

- concerned_shape;
- description;
- element;
- operation;
- quantity;
- role;
- transformation.

4.2.337.1 concerned_shape

The concerned_shape specifies the set of Shape_aspect (see 4.2.394) objects that are affected by the Process_operation (see 4.2.336).

See 4.3.926 for the application assertion.

4.2.337.2 description

The description specifies additional information about the Process_operation_input_or_output.

See 4.3.924 for the application assertion.

The description need not be specified for a particular Process_operation_input_or_output.

If present, there shall be exactly one object that defines the description for a Process_operation_input_or_output.

4.2.337.3 element

The element specifies the element that plays the role of the input or the output for the particular Process_operation (see 4.2.336) in the context of a Process_plan (see 4.2.341).

See 4.3.920, 4.3.921, 4.3.922, and 4.3.923 for the application assertions.

There shall be exactly one object that defines the element for a Process_operation_input_or_output.

4.2.337.4 operation

The operation specifies the Process_operation (see 4.2.336) that the Process_operation_input_or_output is associated to.

See 4.3.925 for the application assertion.

4.2.337.5 quantity

The quantity specifies the quantity of elements that are identified as input or output for a particular Process_operation (see 4.2.336) in the context of a Process_plan (see 4.2.341). In case there is no quantity specified, the default value is one.

The quantity need not be specified for a particular Process_operation_input_or_output.

4.2.337.6 role

The role specifies whether the identified element plays the role of an input or an output for the particular Process_operation (see 4.2.336) in the context of a Process_plan (see 4.2.341).

4.2.337.7 transformation

The transformation specifies the geometrical Transformation (see 4.2.472) between the local coordinate system of the element acting as Process_operation_input_or_output and the reference coordinate system which is either the coordinate system of the reference tool (if present) for the concerned Process_operation (see 4.2.336) or (if no reference tool is present) the coordinate system of the Process_operation itself.

See 4.3.927 for the application assertion.

The transformation need not be specified for a particular Process_operation_input_or_output.

4.2.338 Process_operation_relationship

A Process_operation_relationship is a relationship between two Process_operation (see 4.2.336) objects.

NOTE – The relating Process_operation and the related Process_operation need not belong to the same Process_plan_version (see 4.2.344).

The data associated with a Process_operation_relationship are the following:

- condition;
- description;
- related;
- relating;
- relation_type;
- waiting_time.

4.2.338.1 condition

The condition specifies the condition under which the particular Process_operation_relationship becomes effective.

EXAMPLE 171 – There may be a Process_operation_relationship of type 'series' between a Process_operation (see 4.2.336) representing the manufacturing process of a part and a Process_operation representing the repair of a malproduced part. In this case the Process_operation_relationship may have a condition stating that this series becomes only effective if the output of the manufacturing process has certain properties that make a repairing process necessary.

The condition need not be specified for a particular Process_operation_relationship.

4.2.338.2 description

The description specifies additional information about the Process_operation_relationship.

See 4.3.928 for the application assertion.

The description need not be specified for a particular Process_operation_relationship.

If present, there shall be exactly one object that defines the description for a Process_operation_relationship.

4.2.338.3 related

The related specifies the second of the two Process_operation (see 4.2.336) objects related by a Process_operation_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.929 for the application assertion.

4.2.338.4 relating

The relating specifies the first of the two Process_operation (see 4.2.336) objects related by a Process_operation_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.929 for the application assertion.

4.2.338.5 relation_type

The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:

– 'sequence': The application object defines a relationship where the relating Process_operation (see 4.2.336) shall be completed before the related Process_operation starts;

– 'alternative': The application object defines a relationship where the related Process_operation may be used alternatively instead of the relating Process_operation;

– 'decomposition': The application object defines a relationship where the related Process_operation is one of the components into which the relating Process_operation is broken up.

4.2.338.6 waiting_time

The waiting_time specifies the time which at least shall elapse between the completion of the relating Process_operation (see 4.2.336) and the start of the related Process_operation.

EXAMPLE 172 – The Process_operation 'polishing' must not be started less than 2 hours after the Process_operation 'painting' was finished.

See 4.3.930 for the application assertion.

The waiting_time need not be specified for a particular Process_operation_relationship.

4.2.339 Process_operation_resource

A Process_operation_resource is a resource that is used to perform a Process_operation (see 4.2.336).

EXAMPLE 173 – A hammer is an example for a Process_operation_resource.

NOTE – The concept of a Process_operation_resource shall be used for resources that are not described as an Item (see 4.2.236).

The data associated with a Process_operation_resource are the following:

– name.

The name specifies the word or group of words by which the Process_operation_resource is referred to.

See 4.3.931 for the application assertion.

The name need not be specified for a particular Process_operation_resource.

If present, there shall be exactly one object that defines the name for a Process_operation_resource.

4.2.340 Process_operation_resource_assignment

A Process_operation_resource_assignment is a mechanism to assign a Process_operation_resource (see 4.2.339) to a Process_operation (see 4.2.336).

The data associated with a Process_operation_resource_assignment are the following:

– operation;

– process_duration_for_resource;

– reason;

– reference_tool;

- resource;
- transformation.

4.2.340.1 operation

The operation specifies the Process_operation (see 4.2.336) that the Process_operation_resource (see 4.2.339) is assigned to.

See 4.3.934 for the application assertion.

4.2.340.2 process_duration_for_resource

The process_duration_for_resource specifies the time which elapses between the start of the use of the Process_operation_resource (see 4.2.339) and the completion of the use of the Process_operation_resource within a particular Process_operation (see 4.2.336).

See 4.3.936 for the application assertion.

The process_duration_for_resource need not be specified for a particular Process_operation_resource_assignment.

4.2.340.3 reason

The reason specifies the rationale behind the use of the Process_operation_resource (see 4.2.339) for a particular Process_operation (see 4.2.336).

See 4.3.933 for the application assertion.

The reason need not be specified for a particular Process_operation_resource_assignment.

If present, there shall be exactly one object that defines the reason for a Process_operation_resource_assignment.

4.2.340.4 reference_tool

The reference_tool specifies whether or not the resource identified by the Process_operation_resource_assignment plays the role of the reference tool for the particular Process_operation (see 4.2.336). In case the role is TRUE, all geometric transformations within the scope of the Process_operation are defined with respect to the local coordinate system of the resource identified by the Process_operation_resource_assignment. There must not be more than one resource playing the role of the reference tool for one Process_operation. In case the resource is of type Process_operation_resource (see 4.2.339), the value of the reference_tool shall be FALSE.

4.2.340.5 resource

The resource specifies the resource that is used to perform a Process_operation.

See 4.3.932 and 4.3.935 for the application assertions.

There shall be exactly one object that defines the resource for a Process_operation_resource_-assignment.

4.2.340.6 transformation

The transformation specifies the geometrical Transformation (see 4.2.472) between the local coordinate system of the Process_operation_resource_assignment and the reference coordinate system which is either the coordinate system of the reference tool (if present) for the concerned Process_operation (see 4.2.336) or (if no reference tool is present) the coordinate system of the Process_operation itself. In the case where the identified resource plays the role of the reference tool, no transformation shall be defined.

See 4.3.937 for the application assertion.

The transformation need not be specified for a particular Process_operation_resource_assignment.

4.2.341 Process_plan

A Process_plan is the manufacturing planning information necessary to realize or produce a particular verion of an Item (see 4.2.236).

Each Process_plan may be a Process_plan_version (see 4.2.344).

The data associated with a Process_plan are the following:

- description;
- location;
- name;
- plan_id;
- produced_output.

4.2.341.1 description

The description specifies additional information about the Process_plan.

See 4.3.939 for the application assertion.

The description need not be specified for a particular Process_plan.

If present, there shall be exactly one object that defines the description for a Process_plan.

4.2.341.2 location

The location specifies the location for which the Process_plan is defined.

EXAMPLE 174 – One Process_plan for manufacturing a particular mudguard is defined for a plant in Sweden and another Process_plan for the same part is defined for a plant in France. The attribute

location does not necessarily need to coincide with the attribute location of the associated process operations, since the main location of the Process_plan might be considered to be a plant in Sweden even if one of the process operations is performed at a supplier's site at Norway.

See 4.3.940 for the application assertion.

The location need not be specified for a particular Process_plan.

4.2.341.3 name

The name specifies the word or group of words by which the Process_plan is referred to.

EXAMPLE 175 – 'ABO Engine Cylinder block machining line' is an example for name.

See 4.3.939 for the application assertion.

The name need not be specified for a particular Process_plan.

If present, there shall be exactly one object that defines the name for a Process_plan.

4.2.341.4 plan_id

The plan_id specifies the identifier of the Process_plan that shall be unique within the scope of an organization.

4.2.341.5 produced_output

The produced_output specifies the resulting output of the Process_plan which the Process_plan was defined for.

See 4.3.938 for the application assertion.

The produced_output need not be specified for a particular Process_plan.

4.2.342 Process_plan_operation_assignment

A Process_plan_operation_assignment is the association of a Process_operation (see 4.2.336) and a Process_plan (see 4.2.341) that the particular Process_operation is part of.

NOTE – This mechanism allows the reuse of a Process_operation in the context of more than one Process_plan.

The data associated with a Process_plan_operation_assignment are the following:

- operation;
- plan.

4.2.342.1 operation

The operation specifies the Process_operation (see 4.2.336) which the Process_plan_operation_assignment is assigning to a particular Process_plan (see 4.2.341).

See 4.3.941 for the application assertion.

4.2.342.2 plan

The plan specifies the Process_plan (see 4.2.341) to which the Process_plan_operation_assignment is assigning a Process_operation (see 4.2.336).

See 4.3.942 for the application assertion.

4.2.343 Process_plan_relationship

A Process_plan_relationship is the relationship between two Process_plan (see 4.2.341) objects.

EXAMPLE 176 – If a Part shall be manufactured at two different manufacturing sites, e.g., one at Paris and the other at Bordeaux, there will be two different process plans, that may be related by a Process_plan_relationship.

The data associated with a Process_plan_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.343.1 description

The description specifies additional information about the Process_plan_relationship.

See 4.3.943 for the application assertion.

The description need not be specified for a particular Process_plan_relationship.

If present, there shall be exactly one object that defines the description for a Process_plan_relationship.

4.2.343.2 related

The related specifies the second of the two Process_plan (see 4.2.341) objects related by a Process_plan_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.944 for the application assertion.

4.2.343.3 relating

The relating specifies the first of the two Process_plan (see 4.2.341) objects related by a Process_plan_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.944 for the application assertion.

4.2.343.4 relation_type

The relation_type specifies the meaning of the relationship.

EXAMPLE 177 – 'alternate_location' is an example for relation_type.

4.2.344 Process_plan_version

A Process_plan_version is a particular version of a Process_plan (see 4.2.341).

A Process_plan_version is a type of Process_plan (see 4.2.341).

The data associated with a Process_plan_version are the following:

- version_id.

The version_id specifies the identification of a particular version of a Process_plan (see 4.2.341).

4.2.345 Process_property

A Process_property is a characteristic of a process related object.

A Process_property is a type of Property (see 4.2.361).

Each Process_property is either a General_process_property (see 4.2.206), a Process_cost_property (see 4.2.333), or a Process_quality_property (see 4.2.346).

The data associated with a Process_property are the following:

- described_object.

The described_object specifies the Process_operation (see 4.2.336) or the Process_plan (see 4.2.341) that has the specified property.

See 4.3.945 and 4.3.946 for the application assertions.

There shall be exactly one object that defines the described_object for a Process_property.

4.2.346 Process_quality_property

A Process_quality_property is a property that documents the result of any quality measure process.

A Process_quality_property is a type of Process_property (see 4.2.345).

The data associated with a Process_quality_property are the following:

- description;

- quality_property_value.

4.2.346.1 description

The description specifies additional information about the Process_quality_property.

See 4.3.947 for the application assertion.

The description need not be specified for a particular Process_quality_property.

If present, there shall be exactly one object that defines the description for a Process_quality_property.

4.2.346.2 quality_property_value

The quality_property_value specifies the value of the Process_quality_property.

See 4.3.948 for the application assertion.

4.2.347 Process_state

A Process_state is a view on an in-process-part definition of a particular version of an Item (see 4.2.236). It corresponds to an intermediate state for the considered Process_plan (see 4.2.341). The identifier of a Process_state shall be unique within the context of an Item_version (see 4.2.248) and a Process_plan_version (see 4.2.344).

A Process_state is a type of Design_discipline_item_definition (see 4.2.134).

The data associated with a Process_state are the following:

- related_item_definition.

The related_item_definition specifies the Design_discipline_item_definition (see 4.2.134) that defines the final part that the in process part is a preliminary stage of.

See 4.3.949 for the application assertion.

4.2.348 Product_class

A Product_class is the identification of a set of similar products to be offered to the market.

EXAMPLE 178 – Figure 47 depicts a detail of an example of a structure build up by Product_class and Product_class_hierarchy (see 4.2.349) objects.

NOTE – Product_class is the application object to which the definitions required for configuration control pertain.

The data associated with a Product_class are the following:

- description;
- id;
- level_type;
- name;

- root;
- version_id.

4.2.348.1 description

The description specifies additional information about the Product_class.

See 4.3.950 for the application assertion.

The description need not be specified for a particular Product_class.

If present, there shall be exactly one object that defines the description for a Product_class.

4.2.348.2 id

The id specifies the identifier of the Product_class that shall be unique.

NOTE – The scope of the uniqueness is usually dependent on the form of implementation; it may be e.g., a physical file or data base.

4.2.348.3 level_type

The level_type specifies the level or category of this Product_class in a hierarchical structure of Product_class objects.

EXAMPLE 179 – A high level Product_class (level type='enterprise') may be used to define all Specification_category (see 4.2.416) and Specification objects for all Product_class objects of an enterprise with several brands and companies.

A second level Product_class (level type='platform') may be used to group all products that are based on the same technical concept (platform); these products may belong to different brands.

A third level Product_class (level type='car family') may be used to group all products that have a common base and a fixed set of characteristics (specification categories).

A fourth level Product_class (level type='car type') may represent products that are offered to the market; this level of Product_class may be defined by the marketing department. A set of specifications sufficient to produce the car is associated with its Product_class. Within this association a distinction is made between standard characteristics (e.g. number of doors) and options that have to be chosen (e.g. outside colour) or that may be chosen by the customer (e.g. radio).

4.2.348.4 name

The name specifies the word or group of words by which the Product_class is referred to.

EXAMPLE 180 – '69 Buick Electra', 'Bugatti Royal', and 'Airbus A320' are examples for name.

See 4.3.950 for the application assertion.

The name need not be specified for a particular Product_class.

If present, there shall be exactly one object that defines the name for a Product_class.

4.2.348.5 root

The root specifies the Product_component (see 4.2.350) that is the top level element in a structure of Product_component objects. A Product_class that identifies a Product_component using root is the Product_class one wants to define a bill-of-material for.

NOTE – This entry is the entry into the bill-of-material of the products of that Product_class. The usage of this attribute reflects split into several bill-of-material structures within the organization.

See 4.3.951 for the application assertion.

The root need not be specified for a particular Product_class.

4.2.348.6 version_id

The version_id specifies the identification of a particular version of a Product_class.

The version_id need not be specified for a particular Product_class.

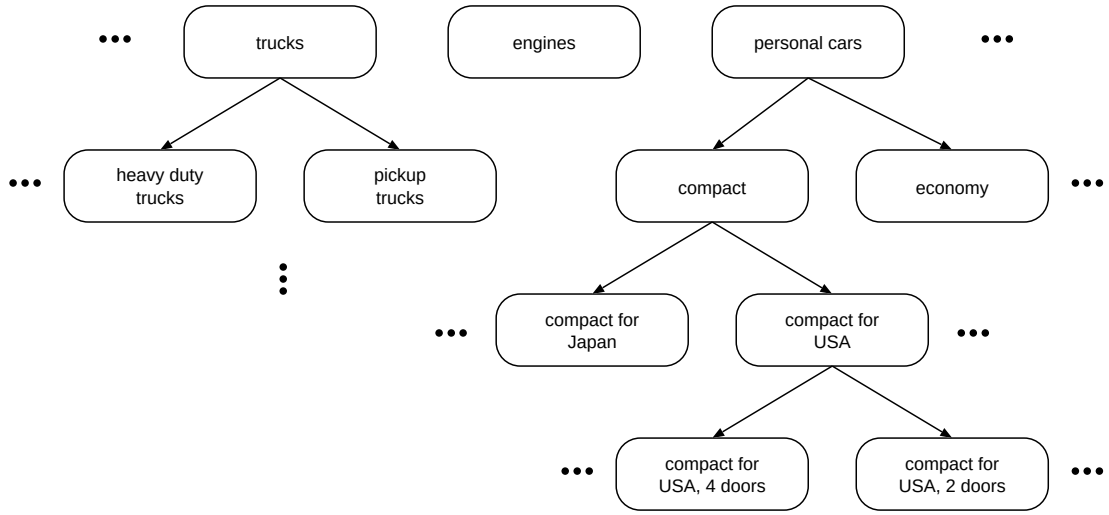


Figure 47 – Example structure of Product_class objects in a hierarchy

4.2.349 Product_class_hierarchy

A Product_class_hierarchy is a hierarchical relationship between two Product_class (see 4.2.348) objects. The set of products of the sub class is included in the set of products of the super class. The hierarchy of Product_class objects defined by the Product_class_hierarchy shall form an acyclic graph. Class_condition_association (see 4.2.65) objects and Class_inclusion_association (see 4.2.67) objects are inherited from the super class to the sub class.

NOTE 1 – At the sub class level other Class_condition_association objects and/or Class_inclusion_association objects may be added.

NOTE 2 – Class_specification_association (see 4.2.68) objects are not inherited but need to be stated explicitly on each level of Product_class objects using the corresponding association type.

The data associated with a Product_class_hierarchy are the following:

- sub_class;
- super_class.

4.2.349.1 sub_class

The sub_class specifies the Product_class (see 4.2.348) in a Product_class_hierarchy that is on a lower level in the hierarchical structure.

See 4.3.952 for the application assertion.

4.2.349.2 super_class

The super_class specifies the Product_class (see 4.2.348) in a Product_class_hierarchy that is on a higher level in the hierarchical structure.

See 4.3.952 for the application assertion.

4.2.350 Product_component

A Product_component is an element in the product decomposition structure. A Product_component is represented by a set of alternate Item_solution (see 4.2.246) objects with common functional requirements. The top level Product_component of the decomposition tree shall be associated to a Product_class (see 4.2.348) as root entry. The corresponding decomposition structure is identical for all variations of all products of that Product_class.

The data associated with a Product_component are the following:

- description;
- id;
- instance_required;
- is_influenced_by;
- is_relevant_for;
- name;
- version_id.

4.2.350.1 description

The description specifies additional information about the Product_component.

See 4.3.955 for the application assertion.

The description need not be specified for a particular Product_component.

If present, there shall be exactly one object that defines the description for a Product_component.

4.2.350.2 id

The id specifies the identifier of the Product_component.

4.2.350.3 instance_required

The instance_required specifies if the existance of a corresponding Item_instance (see 4.2.241) is required for the various Item_solution (see 4.2.246) objects of that Product_component.

NOTE – There are Product_component objects that are not realized by parts . In these cases no instance of an item is required.

EXAMPLE 181 – Trunk space or holes for wire harness are examples for Product_component objects that do not require instances.

4.2.350.4 is_influenced_by

The is_influenced_by specifies the Specification_category (see 4.2.416) objects that impact the design of a solution for the Product_component in the context of the Product_class (see 4.2.348) objects that are referred to by the Class_category_association (see 4.2.64). These Product_class objects shall belong to the set of Product_class objects the Product_component is valid for or shall be subclasses of these Product_class objects.

NOTE – All solutions for the Product_component (i.e. the design of these solutions) are influenced by the referenced Specification_category.

See 4.3.954 for the application assertion.

4.2.350.5 is_relevant_for

The is_relevant_for specifies the Application_context (see 4.2.20) in which the Product_component has to be considered.

See 4.3.953 for the application assertion.

4.2.350.6 name

The name specifies the word or group of words by which the Product_component is referred to.

See 4.3.955 for the application assertion.

The name need not be specified for a particular Product_component.

If present, there shall be exactly one object that defines the name for a Product_component.

4.2.350.7 version_id

The version_id specifies the identification of a particular version of a Product_component.

The version_id need not be specified for a particular Product_component.

4.2.351 Product_component_relationship

A Product_component_relationship is a relationship between two Product_component (see 4.2.350) objects.

The data associated with a Product_component_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.351.1 description

The description specifies additional information about the Product_component_relationship.

See 4.3.956 for the application assertion.

The description need not be specified for a particular Product_component_relationship.

If present, there shall be exactly one object that defines the description for a Product_component_relationship.

4.2.351.2 related

The related specifies the second of the two Product_component (see 4.2.350) objects referenced by the Product_component_relationship.

NOTE – The semantic of this attribute is specified by the attribute relation_type.

See 4.3.957 for the application assertion.

4.2.351.3 relating

The relating specifies the first of the two Product_component (see 4.2.350) objects referenced by the Product_component_relationship.

NOTE – The semantic of this attribute is specified by the attribute relation_type.

See 4.3.957 for the application assertion.

4.2.351.4 relation_type

The relation_type specifies the meaning of the Product_component_relationship. Where applicable the following values shall be used:

- 'decomposition': The application object defines a relationship where the related Product_component (see 4.2.350) is one of the components into which the relating Product_component is broken up.

4.2.352 Product_function

A Product_function is the mode of action or activity by which a product fulfils a certain purpose.

EXAMPLE 182 – The Product_function 'transportation' with subfunctions 'acceleration' and 'retardation' or the Product_function 'illumination' with subfunctions 'exterior illumination' and 'interior illumination' are examples for Product_function objects in a hierarchical structure.

The data associated with a Product_function are the following:

- description;
- id;
- is_relevant_for;
- name;
- version_id.

4.2.352.1 description

The description specifies additional information about the Product_function.

See 4.3.959 for the application assertion.

The description need not be specified for a particular Product_function.

If present, there shall be exactly one object that defines the description for a Product_function.

4.2.352.2 id

The id specifies the identifier of the Product_function.

4.2.352.3 is_relevant_for

The is_relevant_for specifies the Application_context (see 4.2.20) in which the Product_function has to be considered.

See 4.3.958 for the application assertion.

4.2.352.4 name

The name specifies the word or group of words by which the Product_function is referred to.

See 4.3.959 for the application assertion.

The name need not be specified for a particular Product_function.

If present, there shall be exactly one object that defines the name for a Product_function.

4.2.352.5 version_id

The version_id specifies the identification of a particular version of a Product_function.

The version_id need not be specified for a particular Product_function.

4.2.353 Product_function_relationship

A Product_function_relationship is a relationship between two Product_function (see 4.2.352) objects.

NOTE – A Product_function_relationship may be used to build up a hierarchical structure of Product_function objects on the same level of abstraction or to link a Product_function to another Product_function on a lower level of abstraction.

The data associated with a Product_function_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.353.1 description

The description specifies additional information about the Product_function_relationship.

See 4.3.960 for the application assertion.

The description need not be specified for a particular Product_function_relationship.

If present, there shall be exactly one object that defines the description for a Product_function_relationship.

4.2.353.2 related

The related specifies the second of the two Product_function (see 4.2.352) objects related by the Product_function_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.961 for the application assertion.

4.2.353.3 relating

The relating specifies the first of the two Product_function (see 4.2.352) objects related by the Product_function_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.961 for the application assertion.

4.2.353.4 relation_type

The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:

- 'specialization': The application object defines a relationship where the functional requirements for the related Product_function (see 4.2.352) are more specific than those for the relating Product_function;

- 'version sequence': The application object defines a version sequence where the relating Product_function is the preceding version and the related Product_function is the following version.

4.2.354 Product_specification

A Product_specification is the identification of a physically realizable object. It is defined by the Product_class (see 4.2.348) it is a member of and by a set of Specification (see 4.2.415) objects associated to the Product_class. This set of Specification objects shall be sufficient to access the definitional information necessary to manufacture the product, i.e., a Product_specification shall not include possible variations.

The data associated with a Product_specification are the following:

- explicit_product_version;

- name;

- product_class;

- specification;

- version_id.

4.2.354.1 explicit_product_version

The explicit_product_version specifies the Item_version (see 4.2.248) that corresponds to the Product_specification. Both the Product_specification and the Item_version identify the same product.

See 4.3.962 for the application assertion.

The explicit_product_version need not be specified for a particular Product_specification.

4.2.354.2 name

The name specifies the word or group of words by which the Product_specification is referred to.

See 4.3.963 for the application assertion.

The name need not be specified for a particular Product_specification.

If present, there shall be exactly one object that defines the name for a Product_specification.

4.2.354.3 product_class

The product_class specifies the Product_class (see 4.2.348) the Product_specification belongs to.

See 4.3.964 for the application assertion.

4.2.354.4 specification

The specification specifies the set of Specification (see 4.2.415) objects necessary to discriminate the Product_specification within its Product_class (see 4.2.348).

See 4.3.965 for the application assertion.

4.2.354.5 version_id

The version_id specifies the identification of a particular version of a Product_specification.

The version_id need not be specified for a particular Product_specification.

4.2.355 Profile

A Profile is a planar outline used for the definition of features. Each Profile shall lie in the x-y-plane that is either the x-y-plane of its own local coordinate system (in the case where the Profile is used to define a feature that is applied as a Placed_feature (see 4.2.313)) or the x-y-plane defined by the Axis_placement (see 4.2.33) that serves as parameter_reference (in the case where the Profile is used to define a feature that is applied as an Included_feature (see 4.2.232).

Profiles that are swept along a path shall not twist along the path; no twisting in this context means:

- The x-axis of the profile is always parallel to the plane that is the x-z-plane of the 'first occurrence' of the profile.

- The y-axis of the profile is always parallel to the plane that is the y-z-plane of the 'first occurrence' of the profile.

- The z-axis of the profile is always tangent to the path.

Profiles that are swept along a path shall be oriented normal to the path at any position along the path. The geometric transformation placing a profile shall be in a way that the profile plane coincides with the plane normal to the path at the path start or end respectively.

For Profiles that are swept along a path the following rules apply:

If C_1 is the path, C_2 the profile, P any point on C_1, T the tangent of C_1 at P, N the normal of C_1 at P, V the axis defined by P and N, R_c the curvature radius of C_1 at P, C_3 the curve resulting from a) positioning of C_2 in the coordinate space of the feature, b) sweeping of the curve, resulting from the previous step, along C_1 up to P, c) projection of the curve, resulting from the previous step, on the plane defined by P, T, and N, and D_{max} the maximum coordinate of C_3 along Vthen

- N shall exist in any point P of C_1 where $R_c < \infty$;
- $D_{max} \leq R_c$ in any point P of C_1 where $R_c < \infty$.

Each Profile is either an Open_profile (see 4.2.292) or a Closed_profile (see 4.2.71).

The data associated with a Profile are the following:

- is_defined_in;
- parameter_reference.

4.2.355.1 is_defined_in

The is_defined_in specifies the Cartesian_coordinate_space (see 4.2.44) the Profile is defined in. In the case where the Profile is used to define a feature that is applied as a Placed_feature (see 4.2.313) this is the Profile's own local coordinate space. In the case where the Profile is used to define a feature that is applied as an Included_feature (see 4.2.232) this is the part's coordinate space.

See 4.3.967 for the application assertion.

4.2.355.2 parameter_reference

The parameter_reference specifies the Axis_placement (see 4.2.33) to which the parameters of the Profile refer. The parameter_reference shall only be used if the Profile is used to define a feature that is applied as an Included_feature (see 4.2.232). In this case the Axis_placement is defined in the part's coordinate space.

NOTE – In the case of a profile that is used to define a Placed_feature (see 4.2.313) it is the Profile's local coordinate system that serves as reference for the Profile's parameters.

See 4.3.966 for the application assertion.

The parameter_reference need not be specified for a particular Profile.

4.2.356 Project

A Project is a unique process with a time limit, with a defined goal, with a defined budget, and with defined resources.

EXAMPLE 183 – For the development of a new car, a project is set up that is responsible for the development decisions as well as for the accounting of the costs.

The data associated with a Project are the following:

- actual_end_date;
- actual_start_date;
- affected_product_class;
- description;
- id;
- name;
- planned_end_date;
- planned_start_date;
- work_program.

4.2.356.1 actual_end_date

The actual_end_date specifies the date when the Project was actually finished.

See 4.3.969 for the application assertion.

The actual_end_date need not be specified for a particular Project.

4.2.356.2 actual_start_date

The actual_start_date specifies the date when the Project was actually started.

See 4.3.969 for the application assertion.

The actual_start_date need not be specified for a particular Project.

4.2.356.3 affected_product_class

The affected_product_class specifies the Product_class (see 4.2.348) that is affected by the work carried out within a Project.

See 4.3.972 for the application assertion.

4.2.356.4 description

The description specifies additional information about the Project.

See 4.3.971 for the application assertion.

The description need not be specified for a particular Project.

If present, there shall be exactly one object that defines the description for a Project.

4.2.356.5 id

The id specifies the identifier of the Project.

4.2.356.6 name

The name specifies the word or group of words by which the Project is referred to.

See 4.3.971 for the application assertion.

There shall be exactly one object that defines the name for a Project.

4.2.356.7 planned_end_date

The planned_end_date specifies the date when the Project is or was supposed to be finished.

See 4.3.969, 4.3.970, and 4.3.973 for the application assertions.

The planned_end_date need not be specified for a particular Project.

If present, there shall be exactly one object that defines the planned_end_date for a Project.

4.2.356.8 planned_start_date

The planned_start_date specifies the date when the Project is or was supposed to be started.

See 4.3.969 and 4.3.970 for the application assertions.

The planned_start_date need not be specified for a particular Project.

If present, there shall be exactly one object that defines the planned_start_date for a Project.

4.2.356.9 work_program

The work_program specifies the Activity (see 4.2.2) objects that are carried out within the Project.

See 4.3.968 for the application assertion.

4.2.357 Project_relationship

A Project_relationship is a relationship between two Project (see 4.2.356) objects.

The data associated with a Project_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.357.1 description

The description specifies additional information about the Project_relationship.

See 4.3.974 for the application assertion.

The description need not be specified for a particular Project_relationship.

If present, there shall be exactly one object that defines the description for a Project_relationship.

4.2.357.2 related

The related specifies the second of the two Project (see 4.2.356) objects related by a Project_-relationship.

See 4.3.975 for the application assertion.

4.2.357.3 relating

The relating specifies the first of the two Project (see 4.2.356) objects related by a Project_-relationship.

See 4.3.975 for the application assertion.

4.2.357.4 relation_type

The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:

- 'sequence': The application object defines a relationship where the relating Project (see 4.2.356) shall be completed before the related Project starts;
- 'decomposition': The application object defines a relationship where the related Project is one of the components into which the relating Project is broken down;
- 'dependency': The related Project is dependent of the relating Project;
- 'succession': The related Project is the successor of the relating Project.

4.2.358 Projection

A Projection is a condition where a Tolerance_zone (see 4.2.463) is projected from an element of a part. The projection is external to the part and made from one of the ends of the element for a specified length. The concept of projected tolerance zones is defined in ISO 10578.

The data associated with a Projection are the following:

- is_specifying;
- projection_end;
- projection_length.

4.2.358.1 is_specifying

The is_specifying specifies the Tolerance_zone (see 4.2.463) that results from the Projection.

See 4.3.977 for the application assertion.

4.2.358.2 projection_end

The projection_end specifies the Shape_aspect (see 4.2.394) from which the projection is done.

See 4.3.976 for the application assertion.

4.2.358.3 projection_length

The projection_length specifies the distance from the projection_end where the Geometric_-tolerance (see 4.2.216) shall be measured.

4.2.359 Projection_curve

A Projection_curve is the extension of a point, line, surface, or theoretical point of intersection to a location outside the part outline.

A Projection_curve is a type of Annotation_curve (see 4.2.14).

Each Projection_curve may be an Associated_projection_curve (see 4.2.32).

4.2.360 Promissory_usage

A Promissory_usage is a relationship between a constituent of an assembly and the assembly itself. The relationship describes the intention to use the constituent in an assembly. There is no geometric information required for the constituent or the assembly that are associated by a Promissory_usage.

EXAMPLE 184 – In early design stages the Promissory_usage may be used to create preliminary BOMs for prototyping.

A Promissory_usage is a type of Assembly_component_relationship (see 4.2.26).

4.2.361 Property

A Property is an attribute or a characteristic of an item, of a piece of product data, or of a process.

Each Property is either an Item_property (see 4.2.243), a Shape_dependent_property (see 4.2.396), or a Process_property (see 4.2.345).

4.2.362 Property_change

A Property_change is a mechanism to describe the differences between two objects concerning the properties of these objects.

The data associated with a Property_change are the following:

- added_property;
- deleted_property;
- description;
- id;
- is_describing.

4.2.362.1 added_property

The added_property specifies the Property (see 4.2.361) that has been added to the relating object of the relationship specified by the described Change (see 4.2.54).

See 4.3.980 for the application assertion.

4.2.362.2 deleted_property

The deleted_property specifies the Property (see 4.2.361) that has been deleted from the relating object of the relationship specified by the described Change (see 4.2.54).

See 4.3.980 for the application assertion.

4.2.362.3 description

The description specifies additional information about the Property_change.

See 4.3.979 for the application assertion.

There shall be exactly one object that defines the description for a Property_change.

4.2.362.4 id

The id specifies the identifier of the Property_change.

The id need not be specified for a particular Property_change.

4.2.362.5 is_describing

The is_describing specifies the Change (see 4.2.54) that collects the Property_change objects and the Model_change (see 4.2.283) objects.

See 4.3.978 for the application assertion.

4.2.363 Property_relationship

A Property_relationship is a relationship between two Property (see 4.2.361) objects.

The data associated with a Property_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.363.1 description

The description specifies additional information about the Property_relationship.

See 4.3.981 for the application assertion.

The description need not be specified for a particular Property_relationship.

If present, there shall be exactly one object that defines the description for a Property_relationship.

4.2.363.2 related

The related specifies the second of the two Property (see 4.2.361) objects related by the Property_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.982 for the application assertion.

4.2.363.3 relating

The relating specifies the first of the two Property (see 4.2.361) objects related by the Property_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.982 for the application assertion.

4.2.363.4 relation_type

The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:

– 'dependency': The related Property (see 4.2.361) is dependent of the relating Property;

– 'hierachy': The application object defines a hierarchical relationship where the related Property is on a lower level than the relating Property.

4.2.364 Property_value

A Property_value is the representation of a characteristic of a Property (see 4.2.361).

The data associated with a Property_value are the following:

– lower_limit;

– name;

– specified_value;

– upper_limit;

– value_interpretation.

4.2.364.1 lower_limit

The lower_limit specifies the lower limit for the Property_value.

The lower_limit need not be specified for a particular Property_value.

4.2.364.2 name

The name specifies the word or group of words by which the Property_value is referred to.

EXAMPLE 185 – 'maximum length' or 'minimum volume' are examples for the name of a Property_value.

4.2.364.3 specified_value

The specified_value specifies the actual value of a Property_value. The specified_value is either a text description or a numerical measure.

See 4.3.983 and 4.3.984 for the application assertions.

There shall be exactly one object that defines the specified_value for a Property_value.

4.2.364.4 upper_limit

The upper_limit specifies the upper limit for the Property_value.

The upper_limit need not be specified for a particular Property_value.

4.2.364.5 value_interpretation

The value_interpretation specifies information on how the Property_value shall be interpreted.

NOTE – A Property_value may be specified in the design stage of a product but it may also be documented as measured on a prototype.

EXAMPLE 186 – The value_interpretation may be e.g., 'required', 'designed', 'estimated', or 'measured'.

4.2.365 Quantified_instance

A Quantified_instance is the identification of the quantified occurrence of an object that is defined either as a Design_discipline_item_definition (see 4.2.134) or as a Product_specification (see 4.2.354).

EXAMPLE 187 – In a Technical_solution (see 4.2.451) for the front left brake a certain screw is used four times.

A Quantified_instance is a type of Item_instance (see 4.2.241).

The data associated with a Quantified_instance are the following:

– quantity.

The quantity specifies a value and a unit specifying the quantity of occurrences.

NOTE – The quantity may be specified either as a count measure, as a mass measure, as a length measure, as an area measure, or as a volume measure.

See 4.3.985 for the application assertion.

4.2.366 Radiused_bead_end_type

A Radiused_bead_end_type is an end condition of a bead feature that is created by the rotation of the bead profile at the bead start or end about its y-axis.

A Radiused_bead_end_type is a type of Bead_end_type (see 4.2.38).

4.2.367 Radiused_slot_end_type

A Radiused_slot_end_type is an end condition of a slot feature that is created by the rotation of the slot profile at the slot start or end about its y-axis.

NOTE – The resulting radius equals half the width of the swept profile at each position along the y-axis of the swept profile.

EXAMPLE 188 – Figure 48 depicts a Radiused_slot_end_type.

A Radiused_slot_end_type is a type of Slot_end_type (see 4.2.406).

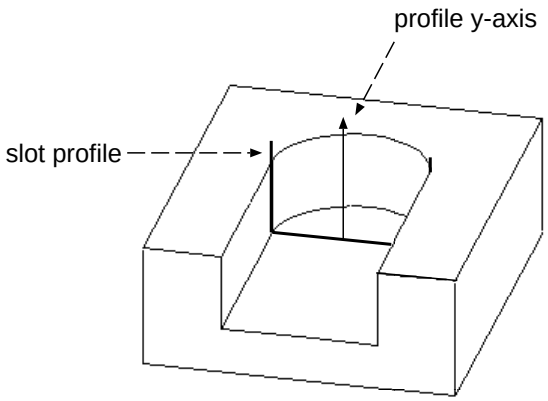


Figure 48 – Radiused_slot_end_type

4.2.368 Rectangular_boss

A Rectangular_boss is a volume adding feature that is specified by an enclosed volume bounded by a rectangular block.

A Rectangular_boss is a type of Boss_feature (see 4.2.42).

The data associated with a Rectangular_boss are the following:

- boundary.

The boundary specifies the Rectangular_closed_profile (see 4.2.370) that defines the cross section of the Rectangular_boss.

See 4.3.986 for the application assertion.

4.2.369 Rectangular_closed_pocket_in_solid

A Rectangular_closed_pocket_in_solid is a volume removal feature whose enclosed volume is characterized by opposite sides being equal in length.

A Rectangular_closed_pocket_in_solid is a type of Pocket_feature_in_solid (see 4.2.321).

The data associated with a Rectangular_closed_pocket_in_solid are the following:

- boundary.

The boundary specifies the Rectangular_closed_profile (see 4.2.370) that defines the cross section of the Rectangular_closed_pocket_in_solid.

See 4.3.987 for the application assertion.

4.2.370 Rectangular_closed_profile

A Rectangular_closed_profile is a closed profile that is specified by an enclosed area bounded by four sides with opposite sides equal in length.

EXAMPLE 189 – Figure 49 depicts a Rectangular_closed_profile.

A Rectangular_closed_profile is a type of Closed_profile (see 4.2.71).

The data associated with a Rectangular_closed_profile are the following:

- corner_radius;
- profile_length;
- profile_width.

4.2.370.1 corner_radius

The corner_radius specifies the size of an arc in all four corners of a Rectangular_closed_profile. For the corner_radius the following constraint applies: $r \leq min(\frac{a}{2}, \frac{b}{2})$ where r is the corner_radius, a is the profile_length, and b is the profile_width.

See 4.3.988 for the application assertion.

The corner_radius need not be specified for a particular Rectangular_closed_profile.

4.2.370.2 profile_length

The profile_length specifies the length of the longest side of a Rectangular_closed_profile. The longest side is oriented along the x-axis.

See 4.3.988 for the application assertion.

4.2.370.3 profile_width

The profile_width specifies the length of the shortest side of a Rectangular_closed_profile. The shortest side is oriented along the y-axis.

See 4.3.988 for the application assertion.

4.2.371 Rectangular_offset_pattern

A Rectangular_offset_pattern is the modification of the placement of a particular occurrence of the base feature in a Rectangular_pattern (see 4.2.374) relative to its expected placement.

EXAMPLE 190 – Figure 50 depicts a Rectangular_offset_pattern.

The data associated with a Rectangular_offset_pattern are the following:

- column_index;
- offset_direction;

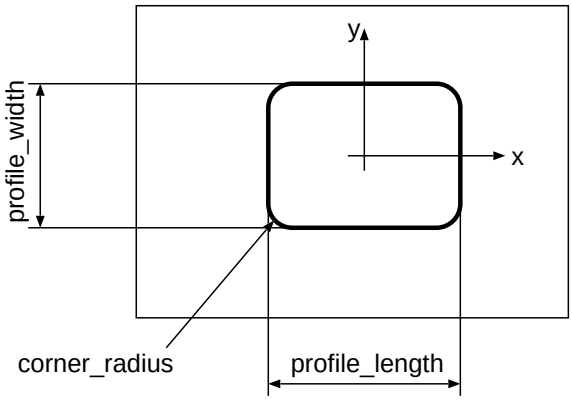


Figure 49 – Rectangular_closed_profile

- offset_distance;
- row_index.

4.2.371.1 column_index

The column_index specifies the index of the column of the Rectangular_offset_pattern.

4.2.371.2 offset_direction

The offset_direction specifies the offset direction of the Rectangular_offset_pattern.

The direction is defined in the same coordinate space as the Rectangular_offset_pattern and lies in a plane that is either parallel to the x-y-plane of this coordinate space (in the case where the Rectangular_offset_pattern is applied as a Placed_feature (see 4.2.313)) or parallel to the x-y-plane of the Axis_placement (see 4.2.33) that serves as the reference (in the casewhere the Rectangular_offset_pattern is applied as an Included_feature (see 4.2.232)).

4.2.371.3 offset_distance

The offset_distance specifies the offset distance of the Rectangular_offset_pattern.

See 4.3.989 for the application assertion.

4.2.371.4 row_index

The row_index specifies the index of the row of the Rectangular_offset_pattern.

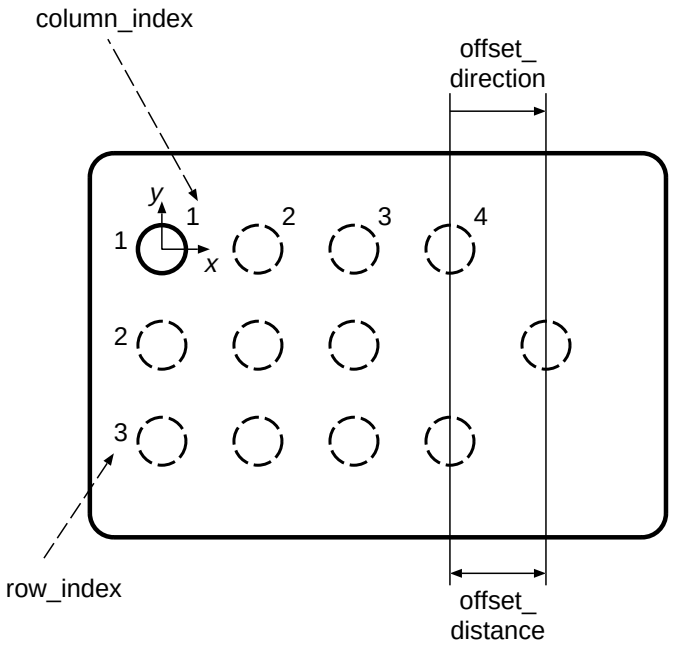


Figure 50 – Rectangular_offset_pattern

4.2.372 Rectangular_omit_pattern

A Rectangular_omit_pattern is an omission of a particular occurrence of the base feature in a Rectangular_pattern (see 4.2.374).

The data associated with a Rectangular_omit_pattern are the following:

- column_index;
- row_index.

4.2.372.1 column_index

The column_index specifies the index of the column of the Rectangular_omit_pattern.

4.2.372.2 row_index

The row_index specifies the index of the row of the Rectangular_omit_pattern.

4.2.373 Rectangular_open_pocket_in_solid

A Rectangular_open_pocket_in_solid is a volume removal feature with opposite sides being equal in length and one side that is not closed.

A Rectangular_open_pocket_in_solid is a type of Pocket_feature_in_solid (see 4.2.321).

The data associated with a Rectangular_open_pocket_in_solid are the following:

- boundary.

The boundary specifies the Square_u_profile (see 4.2.423) that defines the cross section of the Rectangular_open_pocket_in_solid.

See 4.3.990 for the application assertion.

4.2.374 Rectangular_pattern

A Rectangular_pattern is a pattern defined by a base feature arranged in a pattern of rows and columns that lie in the x-y-plane of either the local pattern coordinate space (in the case of a Placed_feature (see 4.2.313)) or of the Axis_placement (see 4.2.33) representing location and orientation of the pattern.

EXAMPLE 191 – Figure 51 illustrates a Rectangular_pattern.

A Rectangular_pattern is a type of Replicate_feature (see 4.2.380).

The data associated with a Rectangular_pattern are the following:

- column_layout_direction;
- column_spacing;
- columns;
- missing_feature;
- relocated_feature;
- row_layout_direction;
- row_spacing;
- rows.

4.2.374.1 column_layout_direction

The column_layout_direction specifies the direction of the column layout of the Rectangular_-pattern.

The direction is defined in the same coordinate space as the Rectangular_pattern and lies in a plane that is either parallel to the x-y-plane of this coordinate space (in the case where the Rectangular_pattern is applied as a Placed_feature (see 4.2.313)) or parallel to the x-y-plane of the Axis_placement (see 4.2.33) that serves as the reference (in the case where the Rectangular_-pattern is applied as an Included_feature (see 4.2.232)).

4.2.374.2 column_spacing

The column_spacing specifies the spacing of the columns of the Rectangular_pattern.

See 4.3.991 for the application assertion.

4.2.374.3 columns

The columns specifies the number of columns of the Rectangular_pattern.

4.2.374.4 missing_feature

The missing_feature specifies those instances that are missing in the Rectangular_pattern.

See 4.3.993 for the application assertion.

4.2.374.5 relocated_feature

The relocated_feature specifies those instances that are relocated in the Rectangular_pattern.

See 4.3.992 for the application assertion.

4.2.374.6 row_layout_direction

The row_layout_direction specifies the direction of the row layout of the Rectangular_pattern.

The direction is defined in the same coordinate space as the Rectangular_pattern and lies in a plane that is either parallel to the x-y-plane of this coordinate space (in the case where the Rectangular_pattern is applied as a Placed_feature (see 4.2.313)) or parallel to the x-y-plane of the Axis_placement (see 4.2.33) that serves as the reference (in the casewhere the Rectangular_-pattern is applied as an Included_feature (see 4.2.232)).

4.2.374.7 row_spacing

The row_spacing specifies the spacing of the rows of the Rectangular_pattern.

See 4.3.991 for the application assertion.

4.2.374.8 rows

The rows specifies the number of rows of the Rectangular_pattern.

4.2.375 Recyclability

A Recyclability is information concerning the ability to reuse objects or components of objects after their primarily intended usage.

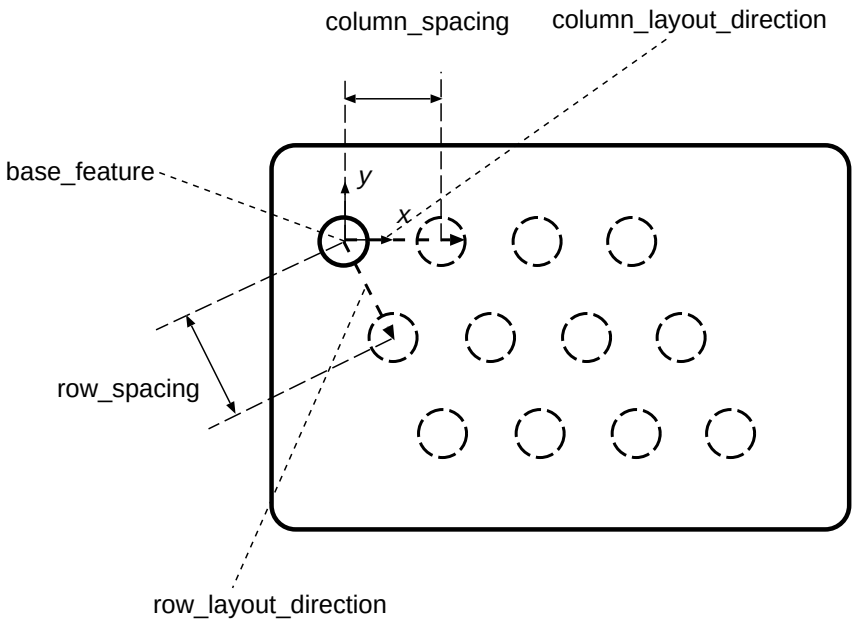


Figure 51 – Rectangular_pattern

A Recyclability is a type of Item_property (see 4.2.243).

The data associated with a Recyclability are the following:

- description;
- recycling_class_value.

4.2.375.1 description

The description specifies additional information about the Recyclability.

See 4.3.994 for the application assertion.

The description need not be specified for a particular Recyclability.

If present, there shall be exactly one object that defines the description for a Recyclability.

4.2.375.2 recycling_class_value

The recycling_class_value specifies the Property_value (see 4.2.364) that represents the value for the Recyclability.

NOTE – The interpretation of this value, e.g., whether the value is required or calculated, is dependent on the semantics of the attribute 'value_interpretation' of the referenced Property_value.

See 4.3.995 for the application assertion.

4.2.376 Reference_plane

A Reference_plane is the definition of a plane that is used to further specify the application of Geometric_tolerance (see 4.2.216) objects. In this plane the two-dimensional tolerance zone is spread from the ideal position of the toleranced element in both directions half the specified tolerance value.

The data associated with a Reference_plane are the following:

- plane_location;
- plane_normal.

4.2.376.1 plane_location

The plane_location specifies a Cartesian point that is defined in the same Cartesian_coordinate_space (see 4.2.44) as the toleranced element and that shall be a point on the plane serving as affected_plane for the Geometric_tolerance (see 4.2.216).

4.2.376.2 plane_normal

The plane_normal specifies a direction vector that is defined in the same Cartesian_coordinate_space (see 4.2.44) as the toleranced element and that indicates a direction normal to the Reference_plane.

4.2.377 Replaced_definition_relationship

A Replaced_definition_relationship is a relationship between two Design_discipline_item_definition (see 4.2.134) objects where the relating Design_discipline_item_definition is replaced by the related Design_discipline_item_definition.

Each Replaced_definition_relationship shall be referenced by an Effectivity (see 4.2.156) that defines the start of the validity of the replacement.

NOTE – A Replaced_definition_relationship may be used to build up version sequences of Design_discipline_item_definition objects.

A Replaced_definition_relationship is a type of Item_definition_relationship (see 4.2.240).

The data associated with a Replaced_definition_relationship are the following:

- description.

The description specifies additional information about the Replaced_definition_relationship.

See 4.3.996 for the application assertion.

The description need not be specified for a particular Replaced_definition_relationship.

If present, there shall be exactly one object that defines the description for a Replaced_definition_relationship.

4.2.378 Replaced_usage_relationship

A Replaced_usage_relationship is a relationship between two Item_instance (see 4.2.241) objects where the relating Item_instance is replaced by the related Item_instance.

Each Replaced_usage_relationship shall be referenced by an Effectivity (see 4.2.156) that defines the start of the validity of the replacement.

A Replaced_usage_relationship is a type of Item_instance_relationship (see 4.2.242).

The data associated with a Replaced_usage_relationship are the following:

- description.

The description specifies additional information about the Replaced_usage_relationship.

See 4.3.997 for the application assertion.

The description need not be specified for a particular Replaced_usage_relationship.

If present, there shall be exactly one object that defines the description for a Replaced_usage_relationship.

4.2.379 Replicate_base_assignment

A Replicate_base_assignment is a mechanism to associate a Replicate_feature (see 4.2.380) with the Feature_definition (see 4.2.175) that is the base feature to be replicated.

The data associated with a Replicate_base_assignment are the following:

- base_feature;
- transformation.

4.2.379.1 base_feature

The base_feature specifies the Feature_definition (see 4.2.175) representing the feature to be replicated.

See 4.3.998 for the application assertion.

4.2.379.2 transformation

The transformation specifies the geometric Transformation (see 4.2.472) used to place the base feature in the Replicate_feature (see 4.2.380)'s coordinate space. In the case of a Replicate_feature applied as a Placed_feature (see 4.2.313) the transformation is used to place the base feature in the local coordinate system of the Replicate_feature.

See 4.3.999 for the application assertion.

The transformation need not be specified for a particular Replicate_base_assignment.

4.2.380 Replicate_feature

A Replicate_feature is a feature that is specified by an existing base feature and the arrangement of identical copies of that feature.

A Replicate_feature is a type of Feature_definition (see 4.2.175).

Each Replicate_feature is either a Circular_pattern (see 4.2.61), a General_pattern (see 4.2.204), or a Rectangular_pattern (see 4.2.374).

The data associated with a Replicate_feature are the following:

- replicate_base.

The replicate_base specifies a Replicate_base_assignment (see 4.2.379) that has the information about the feature that is replicated and possibly about a transformation used to place the base feature in the Replicate_feature' coordinate space.

See 4.3.1000 for the application assertion.

4.2.381 Retention_period

A Retention_period is the definition of a period of time that product data needs to be maintained due to organizational policy or to legal requirements.

The data associated with a Retention_period are the following:

- earliest_end_definition;
- is_applied_to;
- latest_end_definition;
- retention_purpose;
- start_definition.

4.2.381.1 earliest_end_definition

The earliest_end_definition specifies the point in time at which any object the Retention_period is applied to may be deleted. In this context deletion applies to all subordinate objects that are not referenced by other objects.

See 4.3.1010, 4.3.1014, and 4.3.1041 for the application assertions.

There shall be exactly one object that defines the earliest_end_definition for a Retention_period.

4.2.381.2 is_applied_to

The is_applied_to specifies the objects whose existance is controlled by the Retention_period.

NOTE – The master document is the one for which earliest_end_definition and latest_end_definition are the same.

See 4.3.1001, 4.3.1002, 4.3.1003, 4.3.1004, 4.3.1005, 4.3.1006, 4.3.1007, 4.3.1008, 4.3.1009, 4.3.1011, 4.3.1012, 4.3.1013, 4.3.1015, 4.3.1016, 4.3.1017, 4.3.1018, 4.3.1019, 4.3.1020, 4.3.1021, 4.3.1022, 4.3.1023, 4.3.1024, 4.3.1026, 4.3.1027, 4.3.1028, 4.3.1029, 4.3.1030, 4.3.1031, 4.3.1032, 4.3.1033, 4.3.1034, 4.3.1035, 4.3.1036, 4.3.1037, 4.3.1038, 4.3.1039, 4.3.1040, 4.3.1042, and 4.3.1043 for the application assertions.

There shall be at least one object that the Retention_period is_applied_to.

4.2.381.3 latest_end_definition

The latest_end_definition specifies the point in time at which all objects the Retention_period is applied to shall be deleted. In this context deletion applies to all subordinate objects that are not used by other objects.

See 4.3.1010, 4.3.1014, and 4.3.1041 for the application assertions.

There shall be exactly one object that defines the latest_end_definition for a Retention_period.

4.2.381.4 retention_purpose

The retention_purpose specifies the rationale behind the Retention_period.

See 4.3.1025 for the application assertion.

The retention_purpose need not be specified for a particular Retention_period.

If present, there shall be exactly one object that defines the retention_purpose for a Retention_period.

4.2.381.5 start_definition

The start_definition specifies the point in time at which the Retention_period starts.

See 4.3.1010 and 4.3.1014 for the application assertions.

There shall be exactly one object that defines the start_definition for a Retention_period.

4.2.382 Rib_feature

A Rib_feature is a feature that is applied to solid parts to increase the stability of certain regions of a part.

EXAMPLE 192 – Figure 52 illustrates a Rib_feature.

A Rib_feature is a type of Process_feature_in_solid (see 4.2.335).

The data associated with a Rib_feature are the following:

- draft_taper;
- fillet_radius;
- profile_placement;

- rib_profile;
- shoulder_radius;
- sweep_path.

4.2.382.1 draft_taper

The draft_taper specifies a Directed_taper (see 4.2.145) that defines the size of an draft angle that is applied to the side walls of the Rib_feature and the draft direction.

See 4.3.1044 for the application assertion.

The draft_taper need not be specified for a particular Rib_feature.

4.2.382.2 fillet_radius

The fillet_radius specifies a radius shape blend between a Rib_feature and the surrounding faces.

NOTE – If different fillets are required for the various intersection between the Rib_feature and the part Transition_feature (see 4.2.476) objects have to be applied.

See 4.3.1045 for the application assertion.

The fillet_radius need not be specified for a particular Rib_feature.

4.2.382.3 profile_placement

The profile_placement specifies a geometric Transformation (see 4.2.472) that may be used to place the rib_profile in the Rib_feature's coordinate space.

See 4.3.1048 for the application assertion.

The profile_placement need not be specified for a particular Rib_feature.

4.2.382.4 rib_profile

The rib_profile specifies the Open_profile (see 4.2.292) that is swept along the Path (see 4.2.304). The Open_profile shall be shaped in a way that the resulting two sides of the Rib_feature are parallel (before the application of a Directed_taper (see 4.2.145)).

See 4.3.1046 for the application assertion.

4.2.382.5 shoulder_radius

The shoulder_radius specifies the blend radius at the shoulder of the Rib_feature.

See 4.3.1045 for the application assertion.

The shoulder_radius need not be specified for a particular Rib_feature.

4.2.382.6 sweep_path

The sweep_path specifies the Path (see 4.2.304) along which the Open_profile (see 4.2.292) is swept.

See 4.3.1047 for the application assertion.

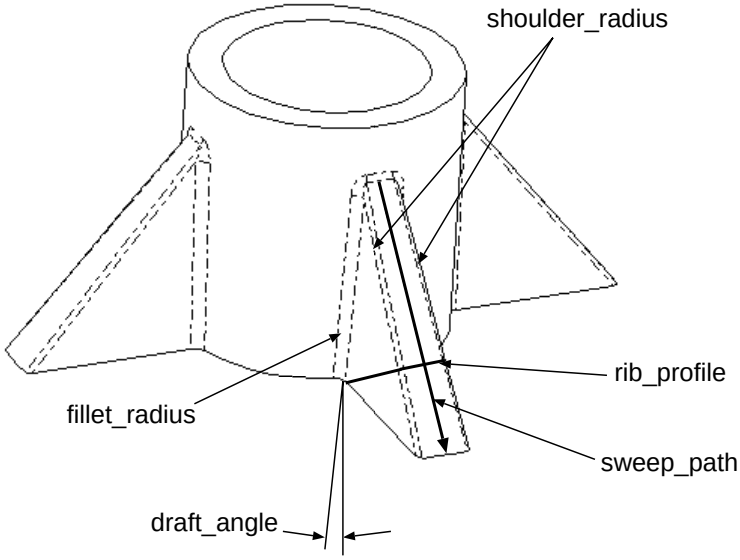


Figure 52 – Rib_feature

4.2.383 Right_angular_wedge

A Right_angular_wedge is a solid that can be envisioned as the result of intersecting a block with a plane perpendicular to one of its faces as shown in figure 53. The received face perpendicular to the cutting plane is triangular or trapezoidal. A Right_angular_wedge is defined by its position in space, by the triangular or trapezoidal face that is defined by three values, and by the depth of the solid.

A Right_angular_wedge is a type of Csg_primitive (see 4.2.101).

4.2.384 Right_circular_cone

A Right_circular_cone is a CSG primitive in the form of a cone that may be truncated. It is defined by an axis, a point on the axis, the semi-angle of the cone, and a distance giving the location in the negative direction along the axis from the point to the base of the cone. In addition, a radius gives the size and location of a truncated face of the cone.

A Right_circular_cone is a type of Csg_primitive (see 4.2.101).

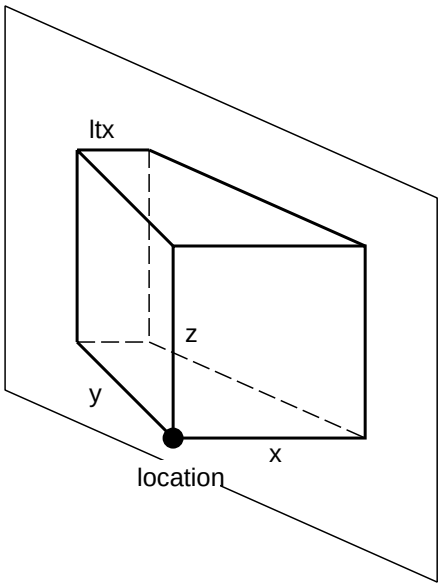


Figure 53 – Right_angular_wedge and its attributes

4.2.385 Right_circular_cylinder

A Right_circular_cylinder is a CSG primitive in the form of a solid cylinder of finite height. It is defined by an axis point at the centre of one planar circular face, an axis, a height, and a radius. The faces are perpendicular to the axis and are circular discs with the specified radius. The height is the distance from the first circular face centre in the positive direction of the axis to the second circular centre.

A Right_circular_cylinder is a type of Csg_primitive (see 4.2.101).

4.2.386 Roughness

A Roughness is a composition of the surface irregularities with relatively small spacing which usually include irregularities resulting from the method of manufacture being used [3]. In the context of this AP Roughness includes information about the method and parameters used to determine these irregularities.

A Roughness is a type of Surface_condition (see 4.2.437).

The data associated with a Roughness are the following:

- cutoff_value;
- measuring_direction;
- measuring_method;
- roughness_max_value;

- roughness_min_value;
- sampling_length.

4.2.386.1 cutoff_value

The cutoff_value specifies the minimal wavelength of the roughness that may be detected in the measuring process.

See 4.3.1050 for the application assertion.

The cutoff_value need not be specified for a particular Roughness.

4.2.386.2 measuring_direction

The measuring_direction specifies the direction in which the Roughness is measured on actual parts.

See 4.3.1049 for the application assertion.

The measuring_direction need not be specified for a particular Roughness.

4.2.386.3 measuring_method

The measuring_method specifies standard and/or the method that is used to determine the Roughness.

NOTE – Roughness values may be specified according to various international standards such as ISO 4287 or ISO 12085.

EXAMPLE 193 – 'Ra - ISO 4287', 'Wy - ISO 4287', 'AR - ISO 12085', or 'W - ISO 12085' are examples for the measuring_method that specify how to interpret the roughness value.

4.2.386.4 roughness_max_value

The roughness_max_value specifies the maximum value with unit for the surface roughness.

See 4.3.1050 for the application assertion.

4.2.386.5 roughness_min_value

The roughness_min_value specifies the minimum value with unit for the surface roughness.

See 4.3.1050 for the application assertion.

The roughness_min_value need not be specified for a particular Roughness.

4.2.386.6 sampling_length

The sampling_length specifies the length interval on a surface within which the surface irregularities are examined.

See 4.3.1050 for the application assertion.

The sampling_length need not be specified for a particular Roughness.

4.2.387 Round_hole

A Round_hole is a hole that is a removal of a volume of cylindrical shape from a solid part.

A Round_hole is a type of Process_feature_in_solid (see 4.2.335).

The data associated with a Round_hole are the following:

- bottom_condition;
- boundary;
- profile_placement;
- sweep_path;
- taper;
- top_chamfer.

4.2.387.1 bottom_condition

The bottom_condition specifies the Hole_bottom_condition (see 4.2.228) that defines the type of the hole bottom.

See 4.3.1056 for the application assertion.

4.2.387.2 boundary

The boundary specifies the Circular_closed_profile (see 4.2.57) that is swept along a Linear_path (see 4.2.268) in order to define the cylinder that is subtracted from the part.

See 4.3.1053 for the application assertion.

4.2.387.3 profile_placement

The profile_placement specifies a geometric Transformation (see 4.2.472) that may be used to place the defining Circular_closed_profile (see 4.2.57) in the Round_hole's coordinate space.

See 4.3.1058 for the application assertion.

The profile_placement need not be specified for a particular Round_hole.

4.2.387.4 sweep_path

The sweep_path specifies the Linear_path (see 4.2.268) that is used as sweep path.

See 4.3.1057 for the application assertion.

4.2.387.5 taper

The taper specifies a possible variation of the diameter of the Round_hole.

See 4.3.1051, 4.3.1054, and 4.3.1055 for the application assertions.

The taper need not be specified for a particular Round_hole.

If present, there shall be exactly one object that defines the taper for a Round_hole.

4.2.387.6 top_chamfer

The top_chamfer specifies a Chamfer_definition (see 4.2.52) that - if present - defines the chamfer at the intersection of Round_hole and part surface.

See 4.3.1052 for the application assertion.

The top_chamfer need not be specified for a particular Round_hole.

4.2.388 Rounded_u_profile

A Rounded_u_profile is an open profile that is specified by a shape bounded by two parallel lines and a semi circle. Each line begins at the end point of the semi circle. The lines are tangent to the circle and extend infinitely.

EXAMPLE 194 – Figure 54 depicts a Rounded_u_profile.

A Rounded_u_profile is a type of Open_profile (see 4.2.292).

The data associated with a Rounded_u_profile are the following:

– width.

The width specifies the distance across a Rounded_u_profile.

See 4.3.1059 for the application assertion.

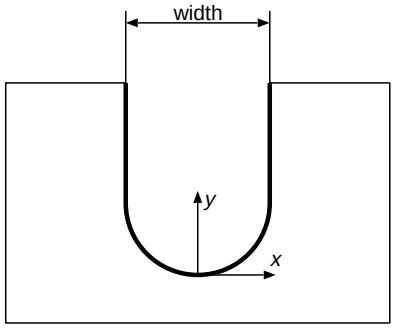


Figure 54 – Rounded_u_profile

4.2.389 Same_time_machining_relationship

A Same_time_machining_relationship is a relationship between two Item_instance (see 4.2.241) objects that specifies that those two objects shall be manufactured together within the same Process_operation (see 4.2.336).

EXAMPLE 195 – Figure 55 shows an example for this case.

A Same_time_machining_relationship is a type of Item_instance_relationship (see 4.2.242).

The data associated with a Same_time_machining_relationship are the following:

- description;
- name;
- transformation.

4.2.389.1 description

The description specifies additional information about the Same_time_machining_relationship.

See 4.3.1061 for the application assertion.

The description need not be specified for a particular Same_time_machining_relationship.

If present, there shall be exactly one object that defines the description for a Same_time_machining_relationship.

4.2.389.2 name

The name specifies the word or group of words by which the Same_time_machining_relationship is referred to.

See 4.3.1061 for the application assertion.

The name need not be specified for a particular Same_time_machining_relationship.

If present, there shall be exactly one object that defines the name for a Same_time_machining_relationship.

4.2.389.3 transformation

The transformation specifies the geometrical Transformation (see 4.2.472) between the local coordinate system of the related Item (see 4.2.236) with respect to the local coordinate system of the relating Item. The transformation is needed to position the related Item with respect to the relating Item for the same time machining process In the case where the transformation is realized by a Template_instance (see 4.2.453) the scale factor of that Template_instance shall be omitted or set to 1.

See 4.3.1060 and 4.3.1062 for the application assertions.

The transformation need not be specified for a particular Same_time_machining_relationship.

If present, there shall be exactly one object that defines the transformation for a Same_time_-machining_relationship.

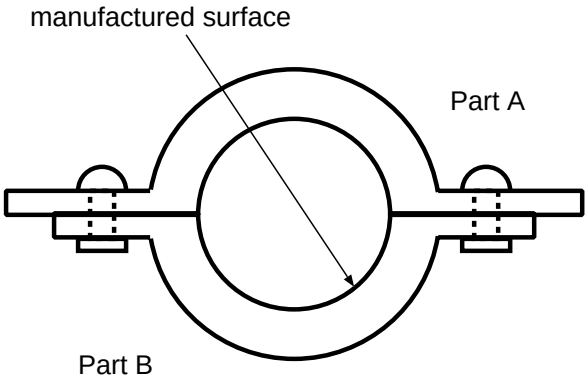


Figure 55 – Same_time_machining_relationship

4.2.390 Security_classification

A Security_classification is the level of confidentiality that is required in order to protect product data against unauthorized usage.

The data associated with a Security_classification are the following:

- is_applied_to;
- name;
- purpose;
- security_classification_level.

4.2.390.1 is_applied_to

The is_applied_to specifies the object with which the Security_classification is associated.

See 4.3.1063, 4.3.1064, 4.3.1065, 4.3.1066, 4.3.1067, 4.3.1068, 4.3.1069, 4.3.1070, 4.3.1071, 4.3.1072, 4.3.1073, 4.3.1074, 4.3.1075, 4.3.1077, 4.3.1078, 4.3.1079, 4.3.1080, 4.3.1081, 4.3.1082, 4.3.1083, 4.3.1084, 4.3.1086, 4.3.1087, 4.3.1088, 4.3.1089, 4.3.1090, 4.3.1091, and 4.3.1092 for the application assertions.

There shall be at least one object that the Security_classification is_applied_to.

4.2.390.2 name

The name specifies the word or group of words used to refer to the Security_classification.

4.2.390.3 purpose

The purpose specifies the rationale behind the Security_classification.

See 4.3.1076 for the application assertion.

There shall be exactly one object that defines the purpose for a Security_classification.

4.2.390.4 security_classification_level

The security_classification_level specifies the Security_level (see 4.2.391) associated with the object.

See 4.3.1085 for the application assertion.

4.2.391 Security_level

A Security_level is the specification of a level of security within some Security_classification (see 4.2.390).

NOTE – The values of Security_level are company specific.

The data associated with a Security_level are the following:

- name;
- used_classification_system.

4.2.391.1 name

The name specifies the word or abbreviation used to refer to the Security_level.

4.2.391.2 used_classification_system

The used_classification_system specifies the Classification_system (see 4.2.70) that contains the information about how to interpret the name of the Security_level.

See 4.3.1093 for the application assertion.

The used_classification_system need not be specified for a particular Security_level.

4.2.392 Selected_instance

A Selected_instance is the instance of an Item (see 4.2.236) whose occurrence depends on certain constraints.

EXAMPLE 196 – To minimize the concentricity error of a wheel weights are attached to the rim; this set of weights is a Selected_instance. The quantity as well as the position of these weights depends on the concentricity behaviour of each particular wheel as manufactured and tested.

A Selected_instance is a type of Item_instance (see 4.2.241).

The data associated with a Selected_instance are the following:

- maximum_quantity;
- minimum_quantity;
- selection_control.

4.2.392.1 maximum_quantity

The maximum_quantity specifies the maximum number of occurrences foreseen for the Selected_-instance.

See 4.3.1094 for the application assertion.

The maximum_quantity need not be specified for a particular Selected_instance.

4.2.392.2 minimum_quantity

The minimum_quantity specifies the minimum number of occurrences foreseen for the Selected_-instance.

See 4.3.1094 for the application assertion.

The minimum_quantity need not be specified for a particular Selected_instance.

4.2.392.3 selection_control

The selection_control specifies the constraint that has to be evaluated for the Selected_instance.

EXAMPLE 197 – 'balancing the wheel' is an example for a selection_control.

4.2.393 Serial_configuration

A Serial_configuration is a Configuration (see 4.2.89) that is planned to apply from a given serial number of the product for whose characteristic the Serial_configuration identifies a solution.

A Serial_configuration is a type of Configuration (see 4.2.89).

The data associated with a Serial_configuration are the following:

- serial_end_number;
- serial_start_number.

4.2.393.1 serial_end_number

The serial_end_number specifies the serial number of that instance of the product that is the last instance for which the Serial_configuration applies.

The serial_end_number need not be specified for a particular Serial_configuration.

4.2.393.2 serial_start_number

The serial_start_number specifies the serial number of that instance of the product that is the first instance for which the Serial_configuration applies.

4.2.394 Shape_aspect

A Shape_aspect is a portion of shape that has to be identified explicitly to be associated with other information.

EXAMPLE 198 – To identify a portion of one face of a shaft that shall have a surface coating applied to it, this portion is specified as a Shape_aspect. This sub-face is not needed to define the geometric shape of the shaft. Figure 56 shows this example.

Each Shape_aspect may be a Feature_occurrence (see 4.2.176).

The data associated with a Shape_aspect are the following:

- composition;
- description.

4.2.394.1 composition

The composition specifies the shape that the Shape_aspect is part of.

See 4.3.1095 for the application assertion.

4.2.394.2 description

The description specifies additional information about the Shape_aspect.

See 4.3.1096 for the application assertion.

The description need not be specified for a particular Shape_aspect.

If present, there shall be exactly one object that defines the description for a Shape_aspect.

4.2.395 Shape_aspect_relationship

A Shape_aspect_relationship is a relationship between two Shape_aspect (see 4.2.394) objects.

The data associated with a Shape_aspect_relationship are the following:

- description;
- related;
- relating;

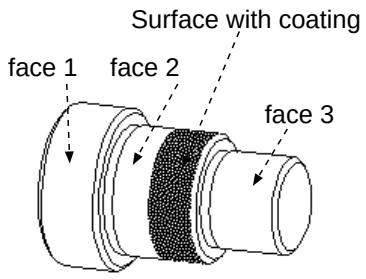


Figure 56 – Shape_aspect on a shaft

– relation_type.

4.2.395.1 description

The description specifies additional information about the Shape_aspect_relationship.

See 4.3.1097 for the application assertion.

The description need not be specified for a particular Shape_aspect_relationship.

If present, there shall be exactly one object that defines the description for a Shape_aspect_relationship.

4.2.395.2 related

The related specifies the second of the two Shape_aspect (see 4.2.394) objects related by a Shape_aspect_relationship.

See 4.3.1098 for the application assertion.

4.2.395.3 relating

The relating specifies the first of the two Shape_aspect (see 4.2.394) objects related by a Shape_aspect_relationship.

See 4.3.1098 for the application assertion.

4.2.395.4 relation_type

The relation_type specifies the meaning of the relationship.

EXAMPLE 199 – 'transition priority' may be a used to define the precedence between two intersecting Transition_feature (see 4.2.476) objects.

4.2.396 Shape_dependent_property

A Shape_dependent_property is a characteristic of an object that is closely related to the shape of the product.

NOTE – In the ideal case most of the Shape_dependent_property objects can be derived from the geometric shape representation. Yet, in many cases the used CAD models are (by intention) not detailed or accurate enough to compute the exact Centre_of_mass (see 4.2.49) or Moments_of_inertia (see 4.2.285) needed for further calculations. Therefore it is necessary to represent these values explicitly.

A Shape_dependent_property is a type of Property (see 4.2.361).

Each Shape_dependent_property is either a Centre_of_mass (see 4.2.49), a Thickness (see 4.2.456), or a Moments_of_inertia (see 4.2.285).

The data associated with a Shape_dependent_property are the following:

– described_object.

The described_object specifies the object that the Shape_dependent_property is associated to.

See 4.3.1099 and 4.3.1100 for the application assertions.

There shall be exactly one object that defines the described_object for a Shape_dependent_property.

4.2.397 Shape_description_association

A Shape_description_association is a mechanism to associate the definition of an Item_shape (see 4.2.245) with the geometric representation of the shape.

The data associated with a Shape_description_association are the following:

– defining_geometry;

– is_defining_shape_for.

4.2.397.1 defining_geometry

The defining_geometry specifies the Geometric_model (see 4.2.211) or the External_model (see 4.2.166) that contains the shape information.

See 4.3.1101 and 4.3.1104 for the application assertions.

There shall be exactly one object that defines the defining_geometry for a Shape_description_association.

4.2.397.2 is_defining_shape_for

The is_defining_shape_for specifies the Item_shape (see 4.2.245) the Shape_description_association is associated with.

See 4.3.1102, 4.3.1103, 4.3.1105, 4.3.1106, 4.3.1107, 4.3.1108, and 4.3.1109 for the application assertions.

There shall be exactly one object that the Shape_description_association is_defining_shape_for.

4.2.398 Shell

A Shell is a collection of Face (see 4.2.167) objects that establish a Surface_model (see 4.2.439).

A Shell is a type of Detailed_geometric_model_element (see 4.2.136).

The data associated with a Shell are the following:

- face.

The face specifies the set of Face (see 4.2.167) objects that form a Shell.

See 4.3.1110 for the application assertion.

4.2.399 Significant_number_of_digits

A Significant_number_of_digits is a limit specification for general tolerance table cells. A Significant_number_of_digits is used for a dimension specification of a part or tool. The number of digits used for the dimension determine which default tolerance entry in the general tolerance table is valid for this particular dimension

The data associated with a Significant_number_of_digits are the following:

- number_of_digits.

The number_of_digits specifies the significant number of digits used for a dimension specification of a part.

4.2.400 Simple_pair_range

A Simple_pair_range is the valid range of the pair parameters of a pair. A Simple_pair_range shall represent either a cylindrical pair range, a gear pair range, a planar pair range, a prismatic pair range, a rack and pinion pair range, a revolute pair range, a rolling curve pair range, a rolling surface pair range, a screw pair range, a sliding curve pair range, a sliding surface pair range, a spherical pair range, or an universal pair range.

The data associated with a Simple_pair_range are the following:

- is_applied_to.

The is_applied_to specifies the Kinematic_pair (see 4.2.258) that the Simple_pair_range applies to.

See 4.3.1111 for the application assertion.

4.2.401 Simulation_information

A Simulation_information is the set of information describing the simulation related properties of an object.

The data associated with a Simulation_information are the following:

- described_object;
- ground;
- mechanism.

4.2.401.1 described_object

The described_object specifies the object to which the Simulation_information belongs.

See 4.3.1112, 4.3.1114, 4.3.1115, 4.3.1116, 4.3.1117, 4.3.1118, 4.3.1120, 4.3.1121, and 4.3.1122 for the application assertions.

There shall be at least one object that defines the described_object for a Simulation_information.

4.2.401.2 ground

The ground specifies the environment with the coordinate space that is the reference for the kinematic data.

NOTE – A geometric description of those parts of a kinematic structure that are immobile may also be part of the ground representation.

See 4.3.1113 for the application assertion.

4.2.401.3 mechanism

The mechanism specifies the Kinematic_mechanism (see 4.2.257) objects that are part of the Simulation_information.

See 4.3.1119 for the application assertion.

4.2.402 Simultaneous_activity

A Simultaneous_activity is a set of two or more Activity (see 4.2.2) objects that shall be carried out simultaneously or shall be implemented together in order to ensure consistency and enhance efficiency.

The data associated with a Simultaneous_activity are the following:

- description;
- element.

4.2.402.1 description

The description specifies additional information about the Simultaneous_activity.

See 4.3.1124 for the application assertion.

The description need not be specified for a particular Simultaneous_activity.

If present, there shall be exactly one object that defines the description for a Simultaneous_activity.

4.2.402.2 element

The element specifies the Activity (see 4.2.2) objects that shall be handled simultaneously.

See 4.3.1123 for the application assertion.

4.2.403 Simultaneous_process_operation

A Simultaneous_process_operation is a set of two or more Process_operation (see 4.2.336) objects that shall be carried out simultaneously.

The data associated with a Simultaneous_process_operation are the following:

- cycle_time;
- element.

4.2.403.1 cycle_time

The cycle_time specifies the period of time within which both Process_operation (see 4.2.336) objects have to take place in order to be defined as 'simultaneous'.

The cycle_time need not be specified for a particular Simultaneous_process_operation.

4.2.403.2 element

The element specifies the Process_operation (see 4.2.336) objects that are performed at the same time.

See 4.3.1125 for the application assertion.

4.2.404 Single_instance

A Single_instance is one particular occurrence of an object that is defined either as a Design_discipline_item_definition (see 4.2.134) or as a Product_specification (see 4.2.354).

NOTE – A Single_instance may or may not be associated with placement information using an Instance_placement (see 4.2.234).

EXAMPLE 200 – In a Technical_solution (see 4.2.451) for the front left brake a brake disk is used that has a defined position specified by a geometric Transformation (see 4.2.472).

A Single_instance is a type of Item_instance (see 4.2.241).

4.2.405 Size_dimension

A Size_dimension is the representation of a dimension characterized as the size of an individual geometric or annotation element.

A Size_dimension is a type of Geometric_dimension (see 4.2.210).

Each Size_dimension may be an Angular_size_dimension (see 4.2.12).

The data associated with a Size_dimension are the following:

- dimension_type;
- is_applied_to.

4.2.405.1 dimension_type

The dimension_type specifies the kind of Size_dimension that is represented.

EXAMPLE 201 – 'diameter size', 'radial size', 'angular size', 'curve size', or 'linear size' are examples for the dimension_type.

4.2.405.2 is_applied_to

The is_applied_to specifies the geometric or annotation element which the Size_dimension is applied to.

See 4.3.1126, 4.3.1127, and 4.3.1128 for the application assertions.

There shall be exactly one object that the Size_dimension is_applied_to.

4.2.406 Slot_end_type

A Slot_end_type is the definition of a kind of end that characterizes a Slot_feature (see 4.2.407).

Each Slot_end_type is either a Radiused_slot_end_type (see 4.2.367), a Flat_slot_end_type (see 4.2.184), an Open_slot_end_type (see 4.2.293), or a Woodruff_slot_end_type (see 4.2.499).

The data associated with a Slot_end_type are the following:

- is_applied_to.

The is_applied_to specifies to which end of the path the Slot_end_type applies. The following values shall be used:

- 'start': The Slot_end_type is applied to that end of the Slot_feature (see 4.2.407) that corresponds to the start of the defining Path (see 4.2.304);
- 'end': The Slot_end_type is applied to that end of the Slot_feature that corresponds to the end of the defining Path.

4.2.407 Slot_feature

A Slot_feature is a channel or depression with a continuous course.

EXAMPLE 202 – Figure 57 illustrates examples of slots.

A Slot_feature is a type of Process_feature_in_solid (see 4.2.335).

The data associated with a Slot_feature are the following:

- end_condition;
- maximum_depth;
- profile_placement;
- slot_profile;
- sweep_path.

4.2.407.1 end_condition

The end_condition specifies the Slot_end_type (see 4.2.406) objects that define the types of the slot's ends. A Slot_feature that is defined with a Complete_circular_path (see 4.2.78) shall have two end conditions of type Open_slot_end_type (see 4.2.293).

See 4.3.1131 for the application assertion.

4.2.407.2 maximum_depth

The maximum_depth specifies some amount of distance from the bottom of a slot to a point that is outside of the slot feature. The maximum_depth places a limitation on the slot depth definition so not to interfere with other features that might be nearby.

NOTE – The maximum_depth is not feature definitional.

The maximum_depth need not be specified for a particular Slot_feature.

4.2.407.3 profile_placement

The profile_placement specifies a geometric Transformation (see 4.2.472) that may be used to place the defining Open_profile (see 4.2.292) in the Slot_feature's coordinate space.

See 4.3.1132 for the application assertion.

The profile_placement need not be specified for a particular Slot_feature.

4.2.407.4 slot_profile

The slot_profile specifies the Open_profile (see 4.2.292) that is swept along the sweep path.

See 4.3.1129 for the application assertion.

4.2.407.5 sweep_path

The sweep_path specifies the Path (see 4.2.304) that serves as a sweep path for a Slot_feature.

See 4.3.1130 for the application assertion.

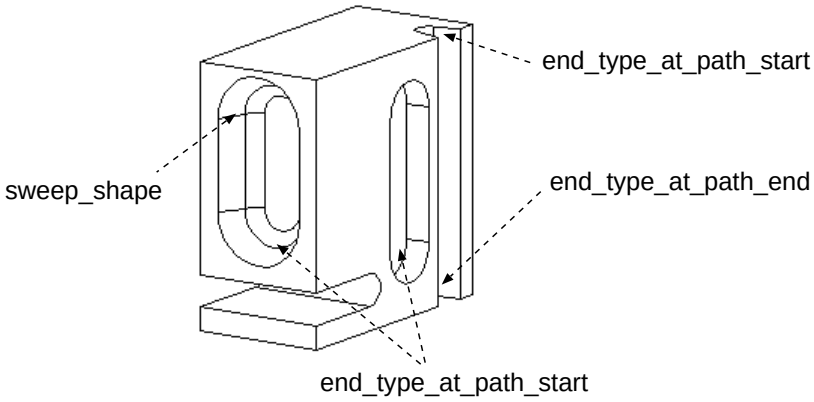


Figure 57 – Slot_feature

4.2.408 Solid_fill_area

A Solid_fill_area is a enclosed area that is uniformly filled by a single colour.

4.2.409 Solid_of_linear_extrusion

A Solid_of_linear_extrusion is a solid that can be formed by sweeping a planar Face (see 4.2.167) along a straight line in space about a specified depth.

A Solid_of_linear_extrusion is a type of Swept_face_solid (see 4.2.443).

4.2.410 Solid_of_revolution

A Solid_of_revolution is a solid formed by revolving a planar Face (see 4.2.167) about an axis. The axis shall be in the plane of the Face and shall not intersect the interior of the Face. The planar Face may have holes which will sweep into holes in the solid. The direction of revolution is clockwise if viewed along the axis in the positive direction.

A Solid_of_revolution is a type of Swept_face_solid (see 4.2.443).

4.2.411 Solid_replica

A Solid_replica is a copy of a solid within a CSG tree at a new location and orientation. Additionally, the received object may have been uniform resized or mirrored.

A Solid_replica is a type of Detailed_geometric_model_element (see 4.2.136).

The data associated with a Solid_replica are the following:

- replicated_solid.

The replicated_solid specifies the solid to be replicated.

See 4.3.1133, 4.3.1134, 4.3.1135, 4.3.1136, 4.3.1137, and 4.3.1138 for the application assertions.

There shall be exactly one object that defines the replicated_solid for a Solid_replica.

4.2.412 Solution_certification

A Solution_certification is a certificate for an Item_solution (see 4.2.246).

The data associated with a Solution_certification are the following:

- certification_date;
- certification_type;
- certified_solution;
- name;
- purpose.

4.2.412.1 certification_date

The certification_date specifies the Date_time (see 4.2.113) the Solution_certification was granted.

See 4.3.1139 for the application assertion.

4.2.412.2 certification_type

The certification_type specifies the kind of certification.

EXAMPLE 203 – 'supplier certificate' is an example for the certification_type.

4.2.412.3 certified_solution

The certified_solution specifies the Item_solution (see 4.2.246) the Solution_certification was granted for.

See 4.3.1140 for the application assertion.

4.2.412.4 name

The name specifies the word or group of words by which the Solution_certification is referred to.

See 4.3.1141 for the application assertion.

There shall be exactly one object that defines the name for a Solution_certification.

4.2.412.5 purpose

The purpose specifies the objective of the Solution_certification.

See 4.3.1141 for the application assertion.

There shall be exactly one object that defines the purpose for a Solution_certification.

4.2.413 Solution_instance_association

A Solution_instance_association is a mechanism to associate an Item_instance (see 4.2.241) as a solution element with an Item_solution (see 4.2.246).

The data associated with a Solution_instance_association are the following:

- instance;
- solution.

4.2.413.1 instance

The instance specifies the Item_instance (see 4.2.241) that is a solution element for the Item_solution (see 4.2.246).

See 4.3.1142 for the application assertion.

4.2.413.2 solution

The solution specifies the Item_solution (see 4.2.246) for which the Item_instance (see 4.2.241) is a solution element.

See 4.3.1143 for the application assertion.

4.2.414 Specific_classification

A Specific_classification is a classification of Item (see 4.2.236) objects according to a specific aspect specified in the classification_type attribute.

A Specific_classification is a type of Item_classification (see 4.2.237).

The data associated with a Specific_classification are the following:

- classification_name;
- classification_type.

4.2.414.1 classification_name

The classification_name specifies the level within the classification type.

EXAMPLE 204 – Within the classification_type 'service item' examples for the classification_name are 'repair' and 'maintenance'.

EXAMPLE 205 – Within the classification_type 'in process item' examples for the classification_name are 'raw material', 'fixing material', and 'final part'.

EXAMPLE 206 – Within the classification_type 'assembly item' examples for the classification_name are 'welding assembly' and 'mounting assembly'.

Within the classification_type 'item role' the following values shall be used where applicable:

– 'part': The item plays the role of a part;

– 'tool': The item plays the role of a tool;

– 'raw material': The item plays the role of raw material.

4.2.414.2 classification_type

The classification_type specifies the kind of classification. Where applicable the following values shall be used:

– 'item role': This type of classification is used for the primary classification of an Item;

– 'assembly item': In this case the attribute 'classification_name' specifies the rationale behind the creation of a specific assembly structure. The Specific_classification shall be applied to the Item (see 4.2.236) whose defining Design_discipline_item_definition (see 4.2.134) is the root of the assembly structure;

– 'completely knocked down item': This type of classification is used to indicate that an Item is used in a production site that has assembling facilities only;

EXAMPLE 207 – The 'completely knocked down' indicates that the decomposition or the delivery condition of an Item may be different as it is described in the BOM used for the serial production. Usually this is the case if the components are shipped to and assembled in a different country.

– 'in process item': This type of classification is used to indicate the state that the Item has in a manufacturing process;

– 'prototype building': This type of classification is used to indicate that an Item is intended for prototype building only; the Item is not intended for serial production;

– 'regulated item': This type of classification is used to indicate that for an Item certain regulations have to be considered;

– 'application control': This type of classification is used to indicate that an Item (see 4.2.236) shall be considered from certification aspects;

EXAMPLE 208 – Before releasing a new car to the market both function and quality of specific parts have to be certified by an authority (e.g. ministry of transportation). For such parts certification requirements have to be considered during the design phase.

– 'safety item': This type of classification is used to indicate that an Item is relevant for safety purposes;

– 'service item': This type of classification is used to indicate that an Item is relevant for service purposes.

NOTE – The classification of an Item as 'standard item' or as 'library item' can be done using the General_classification (see 4.2.192) with a reference to the corresponding Classification_system (see 4.2.70).

4.2.415 Specification

A Specification is a characteristic of a product. A Specification discriminates one product from other constituents of the same Product_class (see 4.2.348). A Specification refers to a Specification_category (see 4.2.416) that completes the semantic of the Specification.

NOTE 1 – A Specification may be a characteristic of the members of more than one Product_class using Class_specification_association (see 4.2.68) objects.

NOTE 2 – A Specification in combination with a Configuration (see 4.2.89) may be used to define under which conditions an Item (see 4.2.236) is used for a product of a Product_class.

EXAMPLE 209 – 'coupe', 'hatchback', or 'sedan' are examples for specifications belonging to the Specification_category of type 'car body'. 'radio 1 with CD player' and 'radio 2 without CD player' are examples for specifications belonging to the Specification_category of type 'radio equipment'. 'disk brake' and 'drum brake' are examples for specifications belonging to the Specification_category of type 'braking system'.

The data associated with a Specification are the following:

– category;

– description;

– id;

– name;

– package;

– version_id.

4.2.415.1 category

The category specifies the Specification_category (see 4.2.416) that completes the semantic of the Specification.

See 4.3.1145 for the application assertion.

4.2.415.2 description

The description specifies additional information about the Specification.

See 4.3.1144 for the application assertion.

The description need not be specified for a particular Specification.

If present, there shall be exactly one object that defines the description for a Specification.

4.2.415.3 id

The id specifies the identifier of the Specification that shall be unique within the scope of a Specification_category (see 4.2.416).

4.2.415.4 name

The name specifies the word or group of words by which the Specification is referred to.

See 4.3.1144 for the application assertion.

The name need not be specified for a particular Specification.

If present, there shall be exactly one object that defines the name for a Specification.

4.2.415.5 package

The package specifies whether this Specification represents a package of Specification objects or not. A package is a set of Specification objects that may be defined by the marketing department. A package combines those Specification objects that shall be offered to the market as a set. In the case where package is 'true' exactly one Specification_inclusion (see 4.2.418) shall refer to this Specification as if_condition.

EXAMPLE 210 – A sports package contains sports seats, a special steering wheel, special tyres and wheels. A winter package contains heated front seats, heated windshield washers, heated exterior mirrors, and special tyres.

NOTE – Usually the members of a package belong to distinct Specification_category (see 4.2.416) objects.

4.2.415.6 version_id

The version_id specifies the identification of a particular version of a Specification.

The version_id need not be specified for a particular Specification.

4.2.416 Specification_category

A Specification_category is the definition of a set of Specification (see 4.2.415) objects for the same purpose.

EXAMPLE 211 – The category 'kind of drive' for the Specification objects 'front wheel drive' and 'rear wheel drive' or the category 'level of equipment' for the Specification objects 'standard', 'comfortable', and 'luxurious' are examples for a Specification_category.

The data associated with a Specification_category are the following:

- description;
- id;
- implicit_exclusive_condition.

4.2.416.1 description

The description specifies information about the Specification_category.

EXAMPLE 212 – The description 'trim' characterizes a Specification_category for Specification (see 4.2.415) objects that controls the quality and colour of the interior parts including e.g., inner door panels and upholstery. Other examples are 'kind of drive', 'rear brake type', 'brake power regulator', 'radio equipment', or 'type of car body'.

See 4.3.1146 for the application assertion.

There shall be exactly one object that defines the description for a Specification_category.

4.2.416.2 id

The id specifies the identifier of the Specification_category that shall be unique.

NOTE – The scope of uniqueness is usually dependent on the form of implementation; it may be e.g., a physical file or a data base.

4.2.416.3 implicit_exclusive_condition

The implicit_exclusive_condition specifies whether the Specification objects within the Specification_category are mutually exclusive for the production of one particular product ('true') or not ('false').

EXAMPLE 213 – 'safety equipment' is an example for a non mutually exclusive Specification_category if e.g. ABS and airbag are Specification objects in this category. 'engine' is an example for a Specification_category with an implicit_exclusive_condition since usually only one engine can be mounted for power supply.

NOTE – More complex conditions can be handled using Specification_expression (see 4.2.417) objects.

4.2.417 Specification_expression

A Specification_expression is a combination of Specification (see 4.2.415) objects formed by Boolean operations.

The data associated with a Specification_expression are the following:

- description;
- operand;
- operation.

4.2.417.1 description

The description specifies additional information about the Specification_expression.

See 4.3.1147 for the application assertion.

The description need not be specified for a particular Specification_expression.

If present, there shall be exactly one object that defines the description for a Specification_expression.

4.2.417.2 operand

The operand specifies the operands of the Boolean operation that are either Specification (see 4.2.415) objects or other Specification_expression objects.

See 4.3.1148 and 4.3.1149 for the application assertions.

There shall be at least one object that defines the operand for a Specification_expression.

4.2.417.3 operation

The operation specifies the kind of Boolean operation. Four kinds of operations are permitted:

- 'and': all of the identified Specification (see 4.2.415) objects shall be used;
- 'or': a subset or all of the identified Specification objects shall be used;
- 'oneof': exactly one of the identified Specification objects shall be used;
- 'not': the identified Specification shall not be used.

4.2.418 Specification_inclusion

A Specification_inclusion is the representation of the statement that the application of a Specification (see 4.2.415) or of a Specification_expression (see 4.2.417) implies the inclusion of an additional Specification or Specification_expression.

NOTE 1 – The Specification_inclusion may apply for a certain Product_class (see 4.2.348) either directly through a Class_inclusion_association (see 4.2.67) or indirectly by inheritance through a Product_class_hierarchy (see 4.2.349).

NOTE 2 – The Specification_inclusion is intended to complete the set of Specification objects for a Product_specification in order to enable the manufacturing of the product on the basis of an initial set of Specification objects defined e.g., by a customer order.

The data associated with a Specification_inclusion are the following:

- description;
- if_condition;
- included_specification.

4.2.418.1 description

The description specifies additional information about the Specification_inclusion.

See 4.3.1150 for the application assertion.

The description need not be specified for a particular Specification_inclusion.

If present, there shall be exactly one object that defines the description for a Specification_-inclusion.

4.2.418.2 if_condition

The if_condition specifies the Specification (see 4.2.415) or the Specification_expression (see 4.2.417) that serves as the condition for the inclusion.

See 4.3.1151 and 4.3.1152 for the application assertions.

There shall be exactly one object that defines the if_condition for a Specification_inclusion.

4.2.418.3 included_specification

The included_specification specifies the Specification (see 4.2.415) or the Specification_expression (see 4.2.417) objects that are to be included.

NOTE 1 – In the case where the included_specification is a Specification_expression that is e.g., an OR expression, there may be several alternatives for Specification to be included.

NOTE 2 – In the case where more than one Specification objects are to be included a Specification_-expression of type AND shall be used.

See 4.3.1151 and 4.3.1152 for the application assertions.

There shall be exactly one object that defines the included_specification for a Specification_-inclusion.

4.2.419 Specified_higher_usage

A Specified_higher_usage is the relationship between a constituent and an assembly where the assembly is not the immediate parent for the constituent. The relationship shows the usage of a constituent in a high level assembly indicating also the path from the high level assembly to the constituent through recursive usage of Specified_higher_usage objects.

A Specified_higher_usage is a type of Assembly_component_relationship (see 4.2.26).

The data associated with a Specified_higher_usage are the following:

- next_usage;
- upper_usage.

4.2.419.1 next_usage

The next_usage specifies the Next_higher_assembly (see 4.2.287) of which the related Item_-instance (see 4.2.241) is a constituent.

NOTE – The next_usage specifies the next node in the path traversing the assembly structure tree from the bottom to the top.

See 4.3.1154 for the application assertion.

4.2.419.2 upper_usage

The upper_usage specifies the Assembly_component_relationship (see 4.2.26) that shows the path from the assembly specified by next_usage to the high level assembly. In case this is a path traversing several nodes, the upper_usage is another Specified_higher_usage object, otherwise it may be a single Next_higher_assembly (see 4.2.287) object.

See 4.3.1153 for the application assertion.

4.2.420 Sphere

A Sphere is a solid of spherical shape defined by a center point and a radius.

A Sphere is a type of Csg_primitive (see 4.2.101).

4.2.421 Spherical_bottom

A Spherical_bottom is a hole bottom that is defined by a hemisphere that is tangent to the sides of the hole Round_hole (see 4.2.387).

A Spherical_bottom is a type of Hole_bottom_condition (see 4.2.228).

The data associated with a Spherical_bottom are the following:

- radius.

The radius specifies the distance between the centre of the sphere and its surface.

See 4.3.1155 for the application assertion.

4.2.422 Spherical_surface

A Spherical_surface is a surface with a constant distance (the radius) from a central point.

A Spherical_surface is a type of Surface (see 4.2.434).

4.2.423 Square_u_profile

A Square_u_profile is an open profile that is bounded by three lines. One is the base line and has a defined length. The other two lines begin at the ends of the base line and extend infinitely at any angle that is equal or larger than a right angle. The corners of a Square_u_profile may be blended by a radius.

EXAMPLE 214 – Figure 58 depicts a Square_u_profile.

A Square_u_profile is a type of Open_profile (see 4.2.292).

The data associated with a Square_u_profile are the following:

- first_angle;
- first_radius;
- second_angle;
- second_radius;
- width.

4.2.423.1 first_angle

The first_angle specifies the size of an angle between the base and that side of the profile that intersects the x-axis at a positive value.

See 4.3.1156 for the application assertion.

4.2.423.2 first_radius

The first_radius specifies a radius shape blend between the base and that side of the profile that intersects the x-axis at a positive value.

See 4.3.1156 for the application assertion.

The first_radius need not be specified for a particular Square_u_profile.

4.2.423.3 second_angle

The second_angle specifies the size of an angle between the base and that side of the profile that intersects the x-axis at a negative value.

See 4.3.1156 for the application assertion.

4.2.423.4 second_radius

The second_radius specifies a radius shape blend between the base and that side of the profile that intersects the x-axis at a negative value.

See 4.3.1156 for the application assertion.

The second_radius need not be specified for a particular Square_u_profile.

4.2.423.5 width

The width specifies the size of the base line for a Square_u_profile.

See 4.3.1156 for the application assertion.

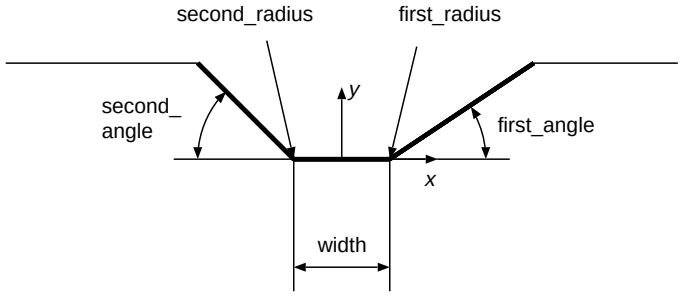


Figure 58 – Square_u_profile

4.2.424 Straightness_tolerance

A Straightness_tolerance is the limitation of the amount of deviation an element of a surface or an axis may have from being a straight line. It defines a tolerance zone within which the considered element or axis must lie. The Straightness_tolerance applies to the complete surface or axis or to each segment of a defined size if the segment attribute is specified.

A Straightness_tolerance is a type of Geometric_tolerance (see 4.2.216).

The data associated with a Straightness_tolerance are the following:

- affected_plane;
- segment_size.

4.2.424.1 affected_plane

The affected_plane specifies the plane a certain type of geometric tolerance applies in. This attribute covers information that is given on drawings implicitly by the drawing plane. If the affected_plane attribute is not specified the tolerance applies in all directions which results in a cylindrical tolerance zone (on drawings indicated by the diameter symbol).

See 4.3.1157 for the application assertion.

The affected_plane need not be specified for a particular Straightness_tolerance.

4.2.424.2 segment_size

The segment_size specifies the length (or the edge length) of a segment if the geometric tolerance is not applied to the total length/area of an axis/surface but on a unit base

EXAMPLE 215 – Figure 59 shows an example for a straightness tolerance callout with indication of a segment_size with a length of 200 mm.

The segment_size need not be specified for a particular Straightness_tolerance.

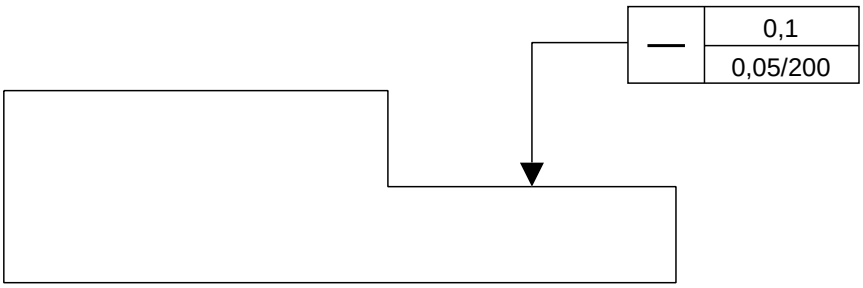


Figure 59 – Indication of segment_size in a tolerance callout

4.2.425 String_with_language

A String_with_language represents text information in a specific language together with an identification of the language used.

The data associated with a String_with_language are the following:

- contents;
- language;
- primary.

4.2.425.1 contents

The contents is text information stored in the language identified by the language attribute.

4.2.425.2 language

The language specifies the language of the text information in the Alpha-3 code specified in ISO 639–2.

EXAMPLE 216 – Possible values for language are 'eng' for English, 'fra' for French, 'rus' for Russian, or 'deu' for German.

4.2.425.3 primary

The primary specifies whether the identified language is the primarily language used or not.

NOTE – In the case where primary is 'false' the text may be a translation.

4.2.426 Structuring_method_hierarchy

A Structuring_method_hierarchy is a mechanism to build up a nesting of Group (see 4.2.222) objects in structuring methods.

NOTE – This mechanism facilitates e.g., the nesting of groups in other groups and the nesting of groups in a layer, but not the nesting of one layer in another layer or group.

The data associated with a Structuring_method_hierarchy are the following:

- related;
- relating.

4.2.426.1 related

The related specifies the Group (see 4.2.222) object to be nested.

See 4.3.1158 for the application assertion.

4.2.426.2 relating

The relating specifies the structuring method in that the Group (see 4.2.222) is nested.

See 4.3.1158 and 4.3.1159 for the application assertions.

There shall be exactly one object that the Structuring_method_hierarchy is relating to.

4.2.427 Styled_element

A Styled_element is a Detailed_element (see 4.2.135) that has presentation information.

A Styled_element is a type of Detailed_element (see 4.2.135).

Each Styled_element is either an Annotation_element (see 4.2.15) or a Styled_model_element (see 4.2.430).

The data associated with a Styled_element are the following:

- appearance;
- invisible.

4.2.427.1 appearance

The appearance specifies the associated presentation information.

See 4.3.1160 for the application assertion.

4.2.427.2 invisible

The invisible specifies if the Styled_element is invisible.

4.2.428 Styled_geometric_model

A Styled_geometric_model is a set of geometric elements that have presentation information.

NOTE – A Styled_geometric_model is defined in the Cartesian_coordinate_space (see 4.2.44) of the corresponding Geometric_model (see 4.2.211).

A Styled_geometric_model is a type of Styled_model (see 4.2.429).

The data associated with a Styled_geometric_model are the following:

- styled_geometry.

The styled_geometry specifies the Geometric_model (see 4.2.211) that contains the styled geometric elements.

See 4.3.1161 for the application assertion.

4.2.429 Styled_model

A Styled_model is a set of geometric or annotation elements that have presentation information.

Each Styled_model may be a Styled_model_version (see 4.2.432).

Each Styled_model is either a Styled_geometric_model (see 4.2.428) or an Annotation_model (see 4.2.16).

The data associated with a Styled_model are the following:

- additional_elements;
- description;
- element;
- model_id.

4.2.429.1 additional_elements

The additional_elements specifies the Template_instance (see 4.2.453) objects that are additionally imported into the Styled_model using a Transformation (see 4.2.472). The additional_elements shall also be of type Styled_model.

See 4.3.1164 for the application assertion.

4.2.429.2 description

The description specifies additional information about the Styled_model.

See 4.3.1162 for the application assertion.

The description need not be specified for a particular Styled_model.

If present, there shall be exactly one object that defines the description for a Styled_model.

4.2.429.3 element

The element specifies the set of Styled_element (see 4.2.427) objects that forms the Styled_model.

See 4.3.1163 for the application assertion.

4.2.429.4 model_id

The model_id specifies the identifier of the Styled_model.

4.2.430 Styled_model_element

A Styled_model_element is a Styled_element (see 4.2.427) that represents geometry.

A Styled_model_element is a type of Styled_element (see 4.2.427).

The data associated with a Styled_model_element are the following:

- element.

The element specifies the Detailed_model_element (see 4.2.137) the presentation information is attached to.

See 4.3.1165 for the application assertion.

4.2.431 Styled_model_relationship

A Styled_model_relationship is a relationship between two Styled_model (see 4.2.429) objects.

The data associated with a Styled_model_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.431.1 description

The description specifies additional information about the Styled_model_relationship.

See 4.3.1166 for the application assertion.

The description need not be specified for a particular Styled_model_relationship.

If present, there shall be exactly one object that defines the description for a Styled_model_relationship.

4.2.431.2 related

The related specifies the second of the two Styled_model (see 4.2.429) objects related by the Styled_model_relationship.

See 4.3.1167 for the application assertion.

4.2.431.3 relating

The relating specifies the first of the two Styled_model (see 4.2.429) objects related by the Styled_model_relationship.

See 4.3.1167 for the application assertion.

4.2.431.4 relation_type

The relation_type specifies the meaning of the relationship.

4.2.432 Styled_model_version

A Styled_model_version is a particular version of a Styled_model (see 4.2.429).

A Styled_model_version is a type of Styled_model (see 4.2.429).

The data associated with a Styled_model_version are the following:

- version_id.

The version_id specifies the identification of a particular version of a Styled_model (see 4.2.429).

4.2.433 Supplier_solution

A Supplier_solution is a solution where the supplier differs from other solutions for the same Product_component (see 4.2.350), for the same Technical_solution (see 4.2.451), or for the same Final_item_solution (see 4.2.182) (as base).

EXAMPLE 217 – The Product_component 'tyre' has a Technical_solution of a '225x50 VR17' type with suppliers A, B, or C. With each of these suppliers a Supplier_solution is associated.

A Supplier_solution is a type of Item_solution (see 4.2.246).

The data associated with a Supplier_solution are the following:

- probability_rate;
- supplier.

4.2.433.1 probability_rate

The probability_rate specifies the share that is assigned to a supplier in the context of the base element.

The probability_rate need not be specified for a particular Supplier_solution.

4.2.433.2 supplier

The supplier specifies the supplier for a Supplier_solution.

4.2.434 Surface

A Surface is a set of mathematical points which is the image of a continuous function defined over a connected subset of the plane R2. It can be envisioned as a set of connected points in 3-dimensional space which is always locally 2-dimensional, but need not be a manifold.

A Surface is a type of Detailed_geometric_model_element (see 4.2.136).

Each Surface is either an Offset_surface (see 4.2.290), a Spherical_surface (see 4.2.422), a Trimmed_surface (see 4.2.479), a B_spline_surface (see 4.2.37), a Surface_of_linear_extrusion (see 4.2.440), a Toroidal_surface (see 4.2.469), a Surface_of_revolution (see 4.2.441), a Plane (see 4.2.317), a Cylindrical_surface (see 4.2.108), or a Conical_surface (see 4.2.91).

4.2.435 Surface_appearance

A Surface_appearance is the definition of the visual appearance of a surface.

The data associated with a Surface_appearance are the following:

- element;
- side.

4.2.435.1 element

The element specifies the elements that belong to a particular Surface_appearance.

EXAMPLE 218 – Surface boundaries, parameter lines, rendering, or fill areas are examples for element.

See 4.3.1168 for the application assertion.

4.2.435.2 side

The side specifies the side of a surface that the Surface_appearance is applied to.

4.2.436 Surface_appearance_element

A Surface_appearance_element is a collection of various elements to be applied in the presentation of the identified side of a surface. The collection includes the appearances for the surface boundaries, parameter lines, rendering, silhouettes, segmentation curves, control grids, and fill areas.

The data associated with a Surface_appearance_element are the following:

- surface_condition.

The surface_condition specifies the surface condition that is presented by the Surface_appearance_element.

See 4.3.1169 for the application assertion.

The surface_condition need not be specified for a particular Surface_appearance_element.

4.2.437 Surface_condition

A Surface_condition is a definition of a characteristic or property of a surface of a part.

Each Surface_condition is either a Visual_appearance (see 4.2.496), a Coating_layer (see 4.2.74), a Tactile_appearance (see 4.2.445), a Treatment_result (see 4.2.477), a Roughness (see 4.2.386), a Contact_ratio (see 4.2.97), or a Hardness (see 4.2.225).

The data associated with a Surface_condition are the following:

- is_applied_to;
- is_caused_by;
- value_interpretation.

4.2.437.1 is_applied_to

The is_applied_to specifies the shape that the Surface_condition is applied to.

See 4.3.1170 and 4.3.1172 for the application assertions.

There shall be at least one object that the Surface_condition is_applied_to.

4.2.437.2 is_caused_by

The is_caused_by specifies the Process_operation (see 4.2.336) that is used to produce the Surface_condition.

See 4.3.1171 for the application assertion.

The is_caused_by need not be specified for a particular Surface_condition.

4.2.437.3 value_interpretation

The value_interpretation specifies information on how the Surface_condition shall be interpreted.

NOTE – A Surface_condition may be specified in the design stage of a product but it may also be documented as measured on a prototype.

EXAMPLE 219 – The value_interpretation may be e.g., 'required', 'designed', 'estimated', or 'measured'.

4.2.438 Surface_condition_symbol

A Surface_condition_symbol is a combination of annotation elements that, as collection, presents a Surface_condition (see 4.2.437).

EXAMPLE 220 – Figure 60 shows examples how a Surface_condition_symbol is presented on technical drawings.

A Surface_condition_symbol is a type of Associated_annotation_symbol (see 4.2.29).

Figure 60 – Presentation of Surface_conditions information on technical drawings

4.2.439 Surface_model

A Surface_model is the representation of geometry formed by a collection of Shell (see 4.2.398) objects.

A Surface_model shall not intersect with another Surface_model except at edges and vertices.

NOTE – Two Surface_model objects may share faces.

A Surface_model is a type of Geometric_model (see 4.2.211).

The data associated with a Surface_model are the following:

- element.

The element specifies the set of Shell (see 4.2.398) objects that form the Surface_model.

See 4.3.1173 for the application assertion.

4.2.440 Surface_of_linear_extrusion

A Surface_of_linear_extrusion is a surface that is defined by sweeping any curve in 3D geometric space in a given direction. In case of a parametric curve, its image shall be swept. The result is an infinite surface that shall be trimmed by applying a Face (see 4.2.167).

A Surface_of_linear_extrusion is a type of Surface (see 4.2.434).

4.2.441 Surface_of_revolution

A Surface_of_revolution is a surface that is defined by rotating any curve, i.e., for a parametric curve its image in three-dimensional geometric space, after a complete revolution about an axis.

A Surface_of_revolution is a type of Surface (see 4.2.434).

4.2.442 Surface_profile_tolerance

A Surface_profile_tolerance is the definition of a uniform boundary along the true profile within which the elements of the surface must lie. The tolerance zone established by a Surface_profile_-tolerance is three dimensional, extending along the length and width of the considered feature as a whole respectively along each segment of a defined size if the segment attribute is specified.

A Surface_profile_tolerance is a type of Geometric_tolerance (see 4.2.216).

The data associated with a Surface_profile_tolerance are the following:

- datum.

The datum specifies the Datum_feature (see 4.2.117) or the Compound_datum (see 4.2.80) that defines the reference for this type of geometric tolerance.

See 4.3.1174 for the application assertion.

4.2.443 Swept_face_solid

A Swept_face_solid is a solid that is defined by a sweep of a planar Face (see 4.2.167). A Swept_face_solid shall not self-intersect.

A Swept_face_solid is a type of Detailed_geometric_model_element (see 4.2.136).

Each Swept_face_solid is either a Solid_of_linear_extrusion (see 4.2.409) or a Solid_of_revolution (see 4.2.410).

The data associated with a Swept_face_solid are the following:

- face.

The face specifies the Face (see 4.2.167) that is swept.

See 4.3.1175 for the application assertion.

4.2.444 Symmetry_tolerance

A Symmetry_tolerance is the amount of deviation a surface may have from being symmetrical to a datum. A plane of the part must lie in a tolerance zone defined by two parallel planes, symmetrically located with respect to a datum plane of the centre.

A Symmetry_tolerance is a type of Geometric_tolerance (see 4.2.216).

The data associated with a Symmetry_tolerance are the following:

- affected_plane;
- datum.

4.2.444.1 affected_plane

The affected_plane specifies the plane a certain type of geometric tolerance applies in. This attribute covers information that is given on drawings implicitly by the drawing plane. If the affected_plane attribute is not specified the tolerance applies in all directions which results in a cylindrical tolerance zone (on drawings indicated by the diameter symbol).

See 4.3.1177 for the application assertion.

The affected_plane need not be specified for a particular Symmetry_tolerance.

4.2.444.2 datum

The datum specifies the Datum_feature (see 4.2.117) or the Compound_datum (see 4.2.80) that defines the reference for this type of geometric tolerance.

See 4.3.1176 for the application assertion.

4.2.445 Tactile_appearance

A Tactile_appearance is the appearance of a surface of a part regarding the sense of touch.

NOTE – At least one of the attributes 'id' or 'name' shall be instantiated.

A Tactile_appearance is a type of Surface_condition (see 4.2.437).

The data associated with a Tactile_appearance are the following:

- depth;
- description;
- id;
- name.

4.2.445.1 depth

The depth specifies the extend of the Tactile_appearance in the normal direction of the surface.

NOTE – The interpretation of the depth depends on the definition of the Tactile_appearance.

See 4.3.1179 for the application assertion.

The depth need not be specified for a particular Tactile_appearance.

4.2.445.2 description

The description specifies additional information about the Tactile_appearance.

See 4.3.1178 for the application assertion.

The description need not be specified for a particular Tactile_appearance.

If present, there shall be exactly one object that defines the description for a Tactile_appearance.

4.2.445.3 id

The id specifies the identifier of the Tactile_appearance.

The id need not be specified for a particular Tactile_appearance.

4.2.445.4 name

The name specifies the word or group of words by which the Tactile_appearance is referred to.

See 4.3.1178 for the application assertion.

The name need not be specified for a particular Tactile_appearance.

If present, there shall be exactly one object that defines the name for a Tactile_appearance.

4.2.446 Target_area

A Target_area is an enclosed area bounded by an arbitrary shape. The shape of the Target_area is described explicitly by a set of curves with reference to the part coordinate system.

A Target_area is a type of Datum_target (see 4.2.118).

The data associated with a Target_area are the following:

- is_defined_by.

The is_defined_by specifies the Shape_aspect (see 4.2.394) that represents the Target_area.

See 4.3.1180 for the application assertion.

4.2.447 Target_circle

A Target_circle is a target formed by an enclosed area bounded by a circle. The origin of the target coordinate system shall be at the centre of the circle. The orientation of the Target_circle is the x-y-plane.

A Target_circle is a type of Placed_target (see 4.2.314).

The data associated with a Target_circle are the following:

- diameter.

The diameter specifies the diameter of the Target_circle.

4.2.448 Target_line

A Target_line is a target formed by a straight curve. The first end point of the Target_line shall be located in the origin of the local target coordinate system. The second end point shall be located on the Z-axis of the local coordinate system at the specified distance.

A Target_line is a type of Placed_target (see 4.2.314).

The data associated with a Target_line are the following:

- target_length.

The target_length specifies the length of the Target_line.

4.2.449 Target_point

A Target_point is a target that is formed by a single point. The Target_point shall be located in the origin of its local coordinate system.

A Target_point is a type of Placed_target (see 4.2.314).

4.2.450 Target_rectangle

A Target_rectangle is a target formed by an enclosed area bounded by four sides with opposite sides equal in length. The origin of the target coordinate system is defined to be at the centre of the rectangle. The orientation of the Target_rectangle is the X-Y-plane with length of the rectangle along the X-axis and the width of the rectangle along the Y-axis of its local coordinate system.

A Target_rectangle is a type of Placed_target (see 4.2.314).

The data associated with a Target_rectangle are the following:

- target_length;
- target_width.

4.2.450.1 target_length

The target_length specifies the length of the Target_rectangle.

4.2.450.2 target_width

The target_width specifies the width of the Target_rectangle.

4.2.451 Technical_solution

A Technical_solution is a solution where the functional requirements are fulfilled in a certain technical way.

EXAMPLE 221 – 'Disc brake', 'disc brake ventilated', 'disc brake with ABS' for Product_component (see 4.2.350) 'front left brake' are examples for a Technical_solution.

A Technical_solution is a type of Item_solution (see 4.2.246).

The data associated with a Technical_solution are the following:

- description.

The description specifies additional information about the Technical_solution.

See 4.3.1181 for the application assertion.

There shall be exactly one object that defines the description for a Technical_solution.

4.2.452 Tee_profile

A Tee_profile is an open profile with a shape similar to the capital letter "T" as shown in figure 61.

A Tee_profile is a type of Open_profile (see 4.2.292).

The data associated with a Tee_profile are the following:

- cross_bar_depth;
- cross_bar_width;
- depth;
- first_angle;
- first_offset;
- radius;
- second_angle;
- second_offset;
- width.

4.2.452.1 cross_bar_depth

The cross_bar_depth specifies the depth dimension of the Tee_profile bar size.

See 4.3.1182 for the application assertion.

4.2.452.2 cross_bar_width

The cross_bar_width specifies the width dimension of the Tee_profile bar size.

See 4.3.1182 for the application assertion.

4.2.452.3 depth

The depth specifies the depth dimension of the Tee_profile stem.

See 4.3.1182 for the application assertion.

4.2.452.4 first_angle

The first_angle specifies the angular measurement for creating a chamfer on the open end of a Tee_profile.

See 4.3.1182 for the application assertion.

4.2.452.5 first_offset

The first_offset specifies the distance from the edge of the Tee_profile stem to create a chamfer on the open end of a Tee_profile.

See 4.3.1182 for the application assertion.

4.2.452.6 radius

The radius specifies an arc size for blending the sides of a Tee_profile cross bar.

See 4.3.1182 for the application assertion.

The radius need not be specified for a particular Tee_profile.

4.2.452.7 second_angle

The second_angle specifies the angular measurement for creating a chamfer between the stem and the cross bar parts of the Tee_profile.

See 4.3.1182 for the application assertion.

4.2.452.8 second_offset

The second_offset specifies the distance from the edge between the Tee_profile stem and the Tee_profile cross bar along the stem to create a chamfer between the stem and the cross bar.

See 4.3.1182 for the application assertion.

4.2.452.9 width

The width specifies the width dimension of the Tee_profile stem.

See 4.3.1182 for the application assertion.

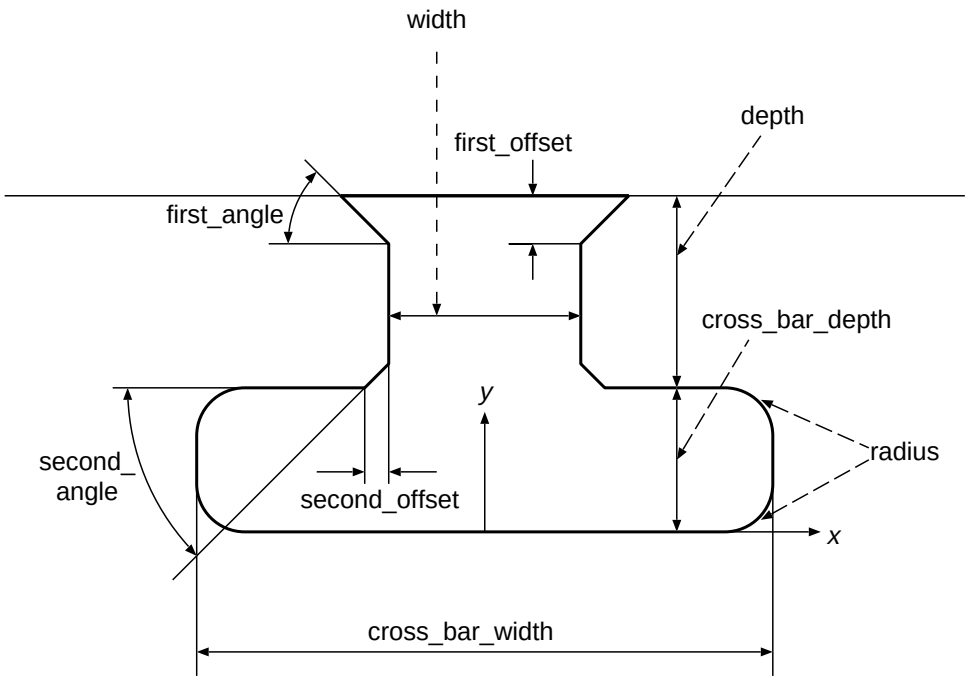


Figure 61 – Tee_profile

4.2.453 Template_instance

A Template_instance is an additional occurrence of a object that has been defined in a different Cartesian_coordinate_space (see 4.2.44) as a Geometric_model (see 4.2.211), a Styled_model (see 4.2.429), or as an External_model (see 4.2.166). A Template_instance is an image copy of this template definition into another Cartesian_coordinate_space where only the location of this copy has to be specified. Additionally uniform scaling, rotation, or mirroring information may be applied to this copy.

NOTE – In the case where the units of the Cartesian coordinate space of the definition are different from the units to be applied to the Template_instance, unit conversion is required. In case of length unit conversion this has to be considered in addition to the scale attribute.

EXAMPLE 222 – In a technical drawing of a mechanical part with several identical drilling holes the hole geometry (circle) together with its annotation elements (diameter dimension and centrelines) is defined once with the name 'annotated drilling hole' and the purpose 'drilling hole representation'. This particular definition is instantiated several times at different locations by corresponding Template_instance objects.

A Template_instance is a type of Detailed_model_element (see 4.2.137).

The data associated with a Template_instance are the following:

- id;
- scale;
- template_definition;

– transformation.

4.2.453.1 id

The id specifies the identifier of the Template_instance.

4.2.453.2 scale

The scale specifies the scaling factor for all Cartesian coordinate directions. The scaling factor shall be positive. If the scaling factor is omitted it shall be 1.0.

The scale need not be specified for a particular Template_instance.

4.2.453.3 template_definition

The template_definition specifies the template to be instantiated.

See 4.3.1183, 4.3.1184, and 4.3.1185 for the application assertions.

There shall be exactly one object that defines the template_definition for a Template_instance.

4.2.453.4 transformation

The transformation specifies the Cartesian transformation applied to the instance. All transformations that can be expressed by an orthonormal 2×2 (for 2D) or 3×3 (for 3D) matrix can be applied, e.g., rotation or mirroring operations.

See 4.3.1186 for the application assertion.

4.2.454 Text

A Text is a collection of characters that convey some human interpretable information.

A Text is a type of Annotation_element (see 4.2.15).

4.2.455 Text_appearance

A Text_appearance is the definition of the visual presentation of text. Text_appearance may be predefined or user defined. Predefined text fonts are the fonts A and B as defined in ISO 3098–1 and ISO 3098–2. User defined text fonts are defined by thickness, colour, text rotation angle, character aspect ratio, character rotation angle, character scale, and character slant angle as shown in figure 62.

4.2.456 Thickness

A Thickness is a geometric property of an object.

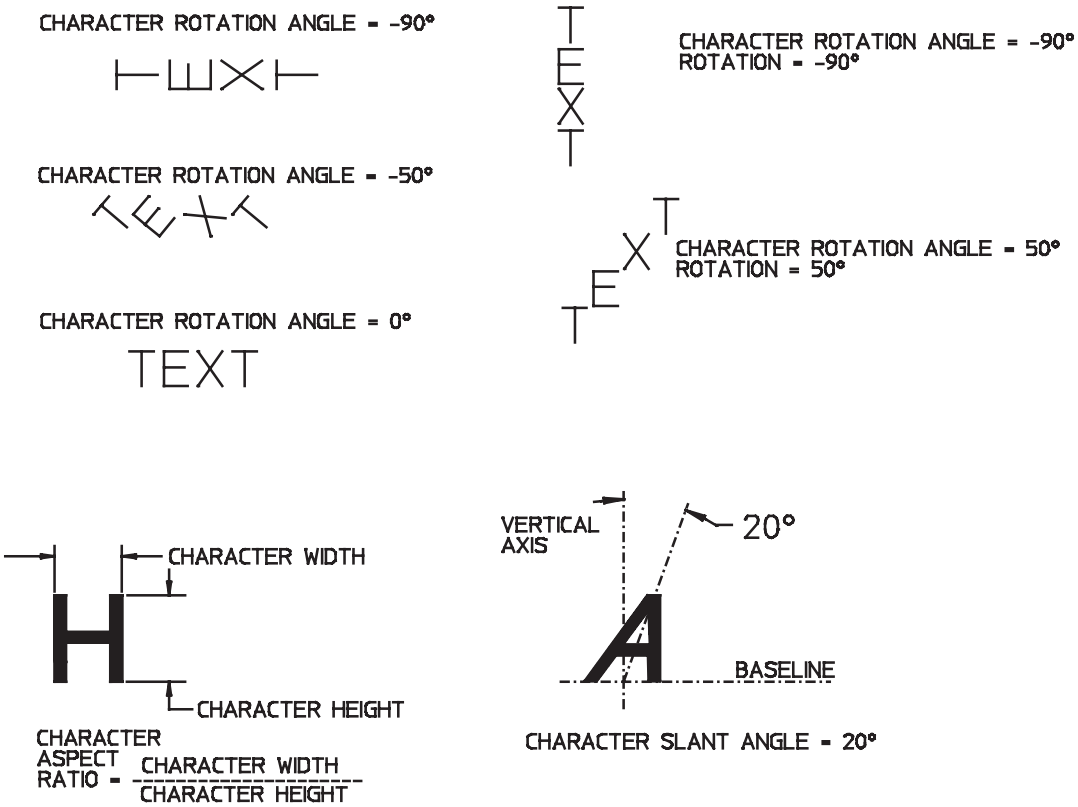


Figure 62 – Text appearance and rotation

A Thickness is a type of Shape_dependent_property (see 4.2.396).

The data associated with a Thickness are the following:

- description;
- material_side;
- thickness_value.

4.2.456.1 description

The description specifies additional information about the Thickness.

See 4.3.1188 for the application assertion.

The description need not be specified for a particular Thickness.

If present, there shall be exactly one object that defines the description for a Thickness.

4.2.456.2 material_side

The material_side specifies that side of a face where the material is.

See 4.3.1187 for the application assertion.

The material_side need not be specified for a particular Thickness.

4.2.456.3 thickness_value

The thickness_value specifies the Property_value (see 4.2.364) that represents the value of the Thickness.

NOTE – The interpretation of this value, e.g., whether the value is required or calculated, is dependent on the semantics of the attribute 'value_interpretation' of the referenced Property_value.

See 4.3.1189 for the application assertion.

4.2.457 Thread_feature

A Thread_feature is a machining feature that is specified by a ridge of uniform section on the form of a helix on the external or internal surface of a cylinder.

EXAMPLE 223 – Thread features may be used to screw parts together.

The data associated with a Thread_feature are the following:

- effective_thread_length;
- inner;
- maximum_thread_length;
- orientation;
- thread_definition.

4.2.457.1 effective_thread_length

The effective_thread_length specifies the length of the thread which is usable by the application.

4.2.457.2 inner

The inner specifies if the thread is applied as an inner thread or as an outer thread.

4.2.457.3 maximum_thread_length

The maximum_thread_length specifies the usable thread length plus the runout.

The maximum_thread_length need not be specified for a particular Thread_feature.

4.2.457.4 orientation

The orientation specifies an Axis_placement (see 4.2.33) whose location represents the start of the thread and whose z-axis points into the direction of the possible runout.

See 4.3.1190 for the application assertion.

4.2.457.5 thread_definition

The thread_definition specifies the Thread_specification (see 4.2.458) that contains the parameters specifying the thread.

See 4.3.1191 for the application assertion.

4.2.458 Thread_specification

A Thread_specification is a set of parameters specifying a thread independent from its application.

Each Thread_specification is either a Defined_thread (see 4.2.127) or a Catalog_thread (see 4.2.48).

The data associated with a Thread_specification are the following:

- qualifier.

The qualifier specifies text information about the thread.

EXAMPLE 224 – 'M 20', 'Tr 40 × 7', 'G ¾', or 'R ½' are examples for the qualifier.

4.2.459 Through_bottom_condition

A Through_bottom_condition is an end condition for a pocket feature that shall pass through two faces of a part.

A Through_bottom_condition is a type of Pocket_bottom_condition (see 4.2.320).

4.2.460 Through_hole_condition

A Through_hole_condition is a condition characterizing a Round_hole (see 4.2.387) that is open at both ends.

A Through_hole_condition is a type of Hole_bottom_condition (see 4.2.228).

4.2.461 Tiling

A Tiling consists of a pattern of tiles that represents graphical symbols defined within a containment border. The containment border defines the edges of the tile. All elements contained within the border are duplicated for each tile. Tiles are placed within a fill area adjacently and do not overlap. The containment border may be blanked. The Tiling may be defined by the user or may be externally defined.

4.2.462 Tolerance_table_cell

A Tolerance_table_cell is the information contained in one cell of a General_tolerance_table (see 4.2.207).

EXAMPLE 225 – Figure 34 depicts an example for a General_tolerance_table.

The data associated with a Tolerance_table_cell are the following:

- cell_value;
- description;
- limit;
- table;
- tolerance_class.

4.2.462.1 cell_value

The cell_value specifies the value of a Tolerance_table_cell.

See 4.3.1194 and 4.3.1197 for the application assertions.

There shall be exactly one object that defines the cell_value for a Tolerance_table_cell.

4.2.462.2 description

The description specifies additional information about the Tolerance_table_cell.

See 4.3.1193 for the application assertion.

The description need not be specified for a particular Tolerance_table_cell.

If present, there shall be exactly one object that defines the description for a Tolerance_table_cell.

4.2.462.3 limit

The limit specifies the limit for the dimensions of a part for which the values given in the Tolerance_table_cell are valid.

See 4.3.1195 and 4.3.1196 for the application assertions.

There shall be exactly one object that defines the limit for a Tolerance_table_cell.

4.2.462.4 table

The table specifies the General_tolerance_table (see 4.2.207) the Tolerance_table_cell belongs to.

See 4.3.1192 for the application assertion.

4.2.462.5 tolerance_class

The tolerance_class specifies the tolerance class for which the values given in the Tolerance_table_cell are valid.

EXAMPLE 226 – Examples for tolerance_class are 'f' for 'fine', 'm' for 'medium', 'c' for 'coarse', and 'v' for 'very coarse'.

4.2.463 Tolerance_zone

A Tolerance_zone is the zone wherein all points of the toleranced geometric element shall fall.

EXAMPLE 227 – A point, a line, a surface, or a middle plane are examples for a toleranced geometric element. Figure 63 depicts a Tolerance_zone of a straightness tolerance of '0.1' specified for a line.

The data associated with a Tolerance_zone are the following:

- form_type;
- is_zone_for.

4.2.463.1 form_type

The form_type specifies the shape of the Tolerance_zone.

EXAMPLE 228 – 'Cylindrical', 'parallelepiped', or 'spherical' are examples for form_type.

4.2.463.2 is_zone_for

The is_zone_for specifies the Geometric_tolerance (see 4.2.216) for that the Tolerance_zone defines the zone.

See 4.3.1198 for the application assertion.

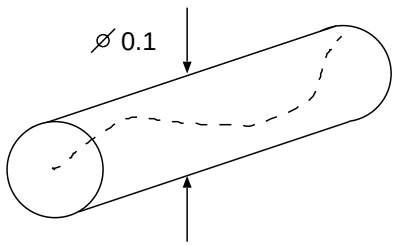


Figure 63 – Tolerance_zone

4.2.464 Tolerance_zone_definition

A Tolerance_zone_definition is the specification of the defining boundaries of a Tolerance_zone (see 4.2.463).

The data associated with a Tolerance_zone_definition are the following:

- first_element;
- is_defining;
- second_element.

4.2.464.1 first_element

The first_element specifies the Shape_aspect (see 4.2.394) that represents the first or - in the case of a cylindrical tolerance zone - the only boundary of the Tolerance_zone_definition.

See 4.3.1199 for the application assertion.

4.2.464.2 is_defining

The is_defining specifies the Tolerance_zone (see 4.2.463) for the Tolerance_zone_definition.

See 4.3.1200 for the application assertion.

4.2.464.3 second_element

The second_element specifies the Shape_aspect (see 4.2.394) that represents the second bounding element.

See 4.3.1199 for the application assertion.

The second_element need not be specified for a particular Tolerance_zone_definition.

4.2.465 Tool_attachment_point_frame

A Tool_attachment_point_frame is a kinematic attachment frame of a tool that may be attached to a link of a Kinematic_mechanism (see 4.2.257).

A Tool_attachment_point_frame is a type of Transformation_to_reference_frame (see 4.2.475).

4.2.466 Tool_center_point_frame

A Tool_center_point_frame is a kinematic reference frame of a tool that may be attached to a link of a Kinematic_mechanism (see 4.2.257).

A Tool_center_point_frame is a type of Transformation_to_reference_frame (see 4.2.475).

4.2.467 Tool_part_relationship

A Tool_part_relationship is a relationship between two Design_discipline_item_definition (see 4.2.134) objects one of which is representing a part (related) and one a tool (relating) that

is used to manufacture the part. The part that is referenced by a Tool_part_relationship is the result of the manufacturing process using the referenced tool.

A Tool_part_relationship is a type of Item_definition_relationship (see 4.2.240).

The data associated with a Tool_part_relationship are the following:

- transformation;
- used_technology_description.

4.2.467.1 transformation

The transformation specifies the relative position of the Item (see 4.2.236) representing the part with respect to the local coordinate system of the Item representing the tool. In the case where the transformation is realized by a Template_instance (see 4.2.453) the scale factor of that Template_instance shall be omitted or set to 1.

See 4.3.1201 and 4.3.1203 for the application assertions.

The transformation need not be specified for a particular Tool_part_relationship.

If present, there shall be exactly one object that defines the transformation for a Tool_part_-relationship.

4.2.467.2 used_technology_description

The used_technology_description specifies the technology used to manufacture the part with the tool and possibly the reasons for the use of a particular technology.

See 4.3.1202 for the application assertion.

The used_technology_description need not be specified for a particular Tool_part_relationship.

If present, there shall be exactly one object that defines the used_technology_description for a Tool_part_relationship.

4.2.468 Topologically_connected_set

A Topologically_connected_set is a collection of Face (see 4.2.167) and/or Oriented_edge_curve (see 4.2.297) objects that form a Compound_model (see 4.2.87).

A Topologically_connected_set is a type of Detailed_geometric_model_element (see 4.2.136).

The data associated with a Topologically_connected_set are the following:

- element.

The element specifies a Face (see 4.2.167) or an Oriented_edge_curve (see 4.2.297) that is part of the Topologically_connected_set.

See 4.3.1204 and 4.3.1205 for the application assertions.

There shall be at least one object that defines the element for a Topologically_connected_set.

4.2.469 Toroidal_surface

A Toroidal_surface is a surface that could be produced by revolving a circle about a non-intersecting line in the plane of its circle. This concept includes cases where the minor radius of the torus is actually greater than the major radius.

A Toroidal_surface is a type of Surface (see 4.2.434).

4.2.470 Torus

A Torus is a solid that is defined by sweeping the area of one circle with minor radius about another circle with larger radius. A Torus is defined by the position in space and the radius of the larger circle, and by the radius of the swept area.

A Torus is a type of Csg_primitive (see 4.2.101).

4.2.471 Total_runout_tolerance

A Total_runout_tolerance is a mechanism to control circular elements of a part to a datum axis the part is rotated about. The tolerance is applied simultaneously to all circular and profile measuring positions as the part is rotated 360 degrees. Where applied to surfaces constructed around a datum axis, Total_runout_tolerance is used to control cumulative variations of circularity, straightness, coaxiality, angularity, taper, and profile of a surface. Where applied to surfaces constructed at right angles to a datum axis, Total_runout_tolerance controls cumulative variations or perpendicularity and flatness.

A Total_runout_tolerance is a type of Geometric_tolerance (see 4.2.216).

The data associated with a Total_runout_tolerance are the following:

- angle;
- datum.

4.2.471.1 angle

The angle specifies the direction a runout tolerance is controlled in. If the angle is specified the runout tolerance applies in this angle which is fixed with respect to the datum axis. If the angle is not specified the runout tolerance applies normal to the feature surface.

The angle need not be specified for a particular Total_runout_tolerance.

4.2.471.2 datum

The datum specifies the Datum_feature (see 4.2.117) or the Compound_datum (see 4.2.80) that defines the reference for this type of geometric tolerance.

See 4.3.1206 for the application assertion.

4.2.472 Transformation

A Transformation is a geometric transformation composed of translation and rotation. Scaling is not included.

Each Transformation is either a Transformation_3d (see 4.2.474) or a Transformation_2d (see 4.2.473).

4.2.473 Transformation_2d

A Transformation_2d is the definition of a geometric transformation in 2D space.

A Transformation_2d is a type of Transformation (see 4.2.472).

4.2.474 Transformation_3d

A Transformation_3d is the definition of a geometric transformation in 3D space.

A Transformation_3d is a type of Transformation (see 4.2.472).

4.2.475 Transformation_to_reference_frame

A Transformation_to_reference_frame is a relative position of one kinematic frame with respect to another kinematic frame.

Each Transformation_to_reference_frame may be one of the following: a Tool_center_point_frame (see 4.2.466) or a Tool_attachment_point_frame (see 4.2.465).

4.2.476 Transition_feature

A Transition_feature is a transition area between two faces. A Transition_feature shall have shape representation.

Each Transition_feature is either an Edge_round (see 4.2.154), a Fillet (see 4.2.181), or a Chamfer (see 4.2.50).

The data associated with a Transition_feature are the following:

- face_1;
- face_2.

4.2.476.1 face_1

The face_1 specifies the first of the two Face (see 4.2.167) objects that the Transition_feature is applied to.

See 4.3.1207 for the application assertion.

4.2.476.2 face_2

The face_2 specifies the second of the two Face (see 4.2.167) objects that the Transition_feature is applied to.

See 4.3.1207 for the application assertion.

4.2.477 Treatment_result

A Treatment_result is a general surface condition that is a result of a certain treatment.

NOTE – At least one of the attributes 'id' or 'name' shall be instantiated.

A Treatment_result is a type of Surface_condition (see 4.2.437).

The data associated with a Treatment_result are the following:

- description;
- id;
- name;
- purpose.

4.2.477.1 description

The description specifies additional information about the Treatment_result.

See 4.3.1208 for the application assertion.

There shall be exactly one object that defines the description for a Treatment_result.

4.2.477.2 id

The id specifies the identifier of the Treatment_result.

The id need not be specified for a particular Treatment_result.

4.2.477.3 name

The name specifies the word or group of words by which the Treatment_result is referred to.

See 4.3.1208 for the application assertion.

The name need not be specified for a particular Treatment_result.

If present, there shall be exactly one object that defines the name for a Treatment_result.

4.2.477.4 purpose

The purpose specifies the effects that are a consequence of the treatment.

See 4.3.1208 for the application assertion.

The purpose need not be specified for a particular Treatment_result.

If present, there shall be exactly one object that defines the purpose for a Treatment_result.

4.2.478 Trimmed_curve

A Trimmed_curve is a curve of finite length that is the result of defining trimming points on an existing curve. The trimming points are the start and end points of the Trimmed_curve. These points may be specified by Cartesian points or by parameter values.

A Trimmed_curve is a type of Curve (see 4.2.102).

4.2.479 Trimmed_surface

A Trimmed_surface is a surface of finite area with identifiable boundaries. A Trimmed_surface is either a b-spline surface, a curve bounded surface, or a rectangular composite surface. A Trimmed_surface shall only be used in Geometrically_bounded_surface_model (see 4.2.220) objects.

A Trimmed_surface is a type of Surface (see 4.2.434).

4.2.480 Upper_lower_limit

An Upper_lower_limit is a limit specification for general tolerance table cells, usually used to discriminate table columns. The range specified by lower_limit and upper_limit corresponds to the dimension specification of a part. The dimensions, usually displayed on a technical drawing, falling into that range have the corresponding tolerance table entry applied if no other specific tolerance value has been assigned to them.

The data associated with an Upper_lower_limit are the following:

- lower_limit;
- upper_limit.

4.2.480.1 lower_limit

The lower_limit specifies the lower value of the value range.

4.2.480.2 upper_limit

The upper_limit specifies the upper value of the value range.

4.2.481 Upper_lower_toleranced_datum

An Upper_lower_toleranced_datum is an entry in a general tolerance table cell that is a tolerance range specified by two values with measure.

The data associated with an Upper_lower_toleranced_datum are the following:

- lower_tolerance_value;
- upper_tolerance_value.

4.2.481.1 lower_tolerance_value

The lower_tolerance_value specifies the lower value with measure of the tolerance range.

4.2.481.2 upper_tolerance_value

The upper_tolerance_value specifies the upper value with measure of the tolerance range.

4.2.482 Value_limit

A Value_limit is one dimension value representing either the lower limit or the upper limit of a Geometric_dimension (see 4.2.210). If a Value_limit is applied to a Geometric_dimension the dimension value shall not be used.

EXAMPLE 229 – '30.5 max' and '5 min' are examples for a Value_limit.

The data associated with a Value_limit are the following:

- limit;
- limit_qualifier.

4.2.482.1 limit

The limit specifies the value of the limit.

4.2.482.2 limit_qualifier

The limit_qualifier specifies the if the limit represents a lower limit ('min') or an upper limit ('max').

4.2.483 Value_range

A Value_range is a pair of dimension values representing the range in which the value of a Geometric_dimension (see 4.2.210) shall lie.

NOTE – This kind of tolerance replaces the nominal value of a Geometric_dimension. All values within the range are considered to be equally good. Usually this way of dimensioning is used for clearances between two aspects of the part rather than for sizes of objects.

The data associated with a Value_range are the following:

- lower_limit;
- upper_limit.

4.2.483.1 lower_limit

The lower_limit specifies the minimum acceptable value of the Geometric_dimension (see 4.2.210) that is constrained by the Value_range.

4.2.483.2 upper_limit

The upper_limit specifies the maximum acceptable value of the Geometric_dimension (see 4.2.210) that is constrained by the Value_range.

4.2.484 Value_termination

A Value_termination specifies a condition where a Joggle_feature (see 4.2.250) has a specified depth.

A Value_termination is a type of Joggle_termination (see 4.2.251).

The data associated with a Value_termination are the following:

- angle;
- depth.

4.2.484.1 angle

The angle specifies the slope angle of the terminating face of the Joggle_feature (see 4.2.250) measured from the panel surface.

See 4.3.1209 for the application assertion.

4.2.484.2 depth

The depth specifies the distance from the rim of the panel to a point that corresponds to the end of the Joggle_feature (see 4.2.250).

See 4.3.1209 for the application assertion.

4.2.485 Value_with_unit

A Value_with_unit is a numerical measure with a corresponding unit.

The data associated with a Value_with_unit are the following:

- unit_component;
- value_component.

4.2.485.1 unit_component

The unit_component specifies the quantity chosen as a standard in terms of which the Value_-with_unit is expressed.

4.2.485.2 value_component

The value_component specifies the quantitative part of a Value_with_unit.

4.2.486 Vector_appearance

A Vector_appearance is the definition of the visual appearance of a vector.

The data associated with a Vector_appearance are the following:

- arrow_head_type;
- arrow_line_font.

4.2.486.1 arrow_head_type

The arrow_head_type specifies the type of the arrow head.

4.2.486.2 arrow_line_font

The arrow_line_font specifies the font of the arrow line. The font determines the visible and invisible segments of the arrow line.

4.2.487 Vee_profile

A Vee_profile is an open profile that is specified by a shape bounded by two lines that connect at a point and extend infinitely. The enclosed angle shall be smaller than 180 degrees. The intersection of the two lines may be blended with a radius.

EXAMPLE 230 – Figure 64 depicts a Vee_profile.

A Vee_profile is a type of Open_profile (see 4.2.292).

The data associated with a Vee_profile are the following:

- profile_angle;
- radius;
- tilt_angle.

4.2.487.1 profile_angle

The profile_angle specifies the size of the angle between the two sides of the Vee_profile. The angle shall be greater than 0 degrees and less than 180 degrees.

See 4.3.1210 for the application assertion.

4.2.487.2 radius

The radius specifies the size of the blend radius at the point of the V, or where the two sides come together.

See 4.3.1210 for the application assertion.

The radius need not be specified for a particular Vee_profile.

4.2.487.3 tilt_angle

The tilt_angle specifies the size of the angle between one side of the Vee_profile and the x-vector of the local coordinate system of the Vee_profile.

See 4.3.1210 for the application assertion.

4.2.488 Vertex_loop

A Vertex_loop is a loop that consists of only one vertex. The dimensionality of a Vertex_loop is zero.

EXAMPLE 231 – The tip of a cone is an example for the usage of a Vertex_loop.

The data associated with a Vertex_loop are the following:

- loop_vertex.

The loop_vertex specifies a Vertex_point (see 4.2.489) that represents the geometry of the associated vertex.

See 4.3.1211 for the application assertion.

4.2.489 Vertex_point

A Vertex_point is a topological construct corresponding to a point. It has dimensionality 0 and extent 0. The geometric location of the Vertex_point is defined by an associated point.

The data associated with a Vertex_point are the following:

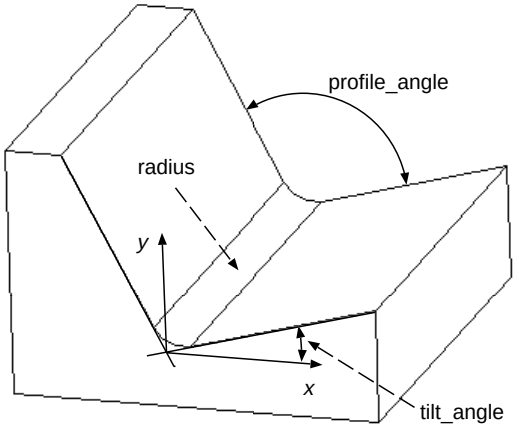


Figure 64 – Vee_profile

– vertex_geometry.

The vertex_geometry specifies the geometric point that defines the position in geometric space of the vertex point.

See 4.3.1212 for the application assertion.

4.2.490 View

A View is the representation of a two-dimensional picture. This picture consists of the image generated by the projection mechanism defined by one Model_image (see 4.2.284) and additional View_placed_annotation (see 4.2.494). A View is placed in the coordinate space of a Drawing_-sheet (see 4.2.152) or of a View_area (see 4.2.491).

The data associated with a View are the following:

– description;

– id;

– layout;

– name;

– transformation;

– view_space.

4.2.490.1 description

The description specifies additional information about the View.

See 4.3.1215 for the application assertion.

The description need not be specified for a particular View.

If present, there shall be exactly one object that defines the description for a View.

4.2.490.2 id

The id specifies the identification of the View.

The id need not be specified for a particular View.

4.2.490.3 layout

The layout references either a Drawing_sheet (see 4.2.152) or a View_area (see 4.2.491) where the View is placed.

See 4.3.1214 and 4.3.1217 for the application assertions.

There shall be exactly one object that defines the layout for a View.

4.2.490.4 name

The name specifies the word or group of words by which the View is referred to.

See 4.3.1215 for the application assertion.

There shall be exactly one object that defines the name for a View.

4.2.490.5 transformation

The transformation specifies the Transformation_2d (see 4.2.473) that is used to place and orient the View on a Drawing_sheet (see 4.2.152) or on a View_area (see 4.2.491).

See 4.3.1216 for the application assertion.

4.2.490.6 view_space

The view_space specifies the two-dimensional Cartesian coordinate space where the presented picture is defined.

See 4.3.1213 for the application assertion.

4.2.491 View_area

A View_area is a collection of View (see 4.2.490) objects placed in a specified rectangular area. The placement of the views is defined relative to the two-dimensional coordinate space of the View_area.

The data associated with a View_area are the following:

- area_space;
- description;
- id;
- name;
- revision_id;
- size.

4.2.491.1 area_space

The area_space specifies the two-dimensional Cartesian coordinate space relative to that the presented views are placed.

See 4.3.1218 for the application assertion.

4.2.491.2 description

The description specifies additional information about the View_area.

See 4.3.1219 for the application assertion.

The description need not be specified for a particular View_area.

If present, there shall be exactly one object that defines the description for a View_area.

4.2.491.3 id

The id specifies the identifier of the View_area.

The id need not be specified for a particular View_area.

4.2.491.4 name

The name specifies the word or group of words by which the View_area is referred to.

See 4.3.1219 for the application assertion.

The name need not be specified for a particular View_area.

If present, there shall be exactly one object that defines the name for a View_area.

4.2.491.5 revision_id

The revision_id specifies the identifier of the revision of the View_area.

The revision_id need not be specified for a particular View_area.

4.2.491.6 size

The size specifies the size of the View_area.

4.2.492 View_area_assignment

A View_area_assignment is a mechanism to associate a View_area (see 4.2.491) with the object that is presented on the View_area.

The data associated with a View_area_assignment are the following:

- is_describing;
- view_area.

4.2.492.1 is_describing

The is_describing specifies the object that is associated with the View_area (see 4.2.491).

See 4.3.1220, 4.3.1221, 4.3.1222, 4.3.1223, 4.3.1224, 4.3.1225, 4.3.1226, 4.3.1227, 4.3.1228, 4.3.1229, 4.3.1230, 4.3.1231, 4.3.1232, 4.3.1233, and 4.3.1234 for the application assertions.

There shall be exactly one object that the View_area_assignment is_describing.

4.2.492.2 view_area

The view_area specifies the View_area (see 4.2.491) that is associated with the object.

See 4.3.1235 for the application assertion.

4.2.493 View_area_relationship

A View_area_relationship is a relationship between two View_area (see 4.2.491) objects.

The data associated with a View_area_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.493.1 description

The description specifies additional information about the View_area_relationship.

See 4.3.1236 for the application assertion.

The description need not be specified for a particular View_area_relationship.

If present, there shall be exactly one object that defines the description for a View_area_relationship.

4.2.493.2 related

The related specifies the secondary view area in the relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.1237 for the application assertion.

4.2.493.3 relating

The relating specifies the primary view area in the relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.1237 for the application assertion.

4.2.493.4 relation_type

The relation_type specifies the kind of relationship. Where applicable the following values shall be used:

- 'copy': The application object defines a relationship where the related View_area (see 4.2.491) is a copy of the relating View_area;

- 'derivation': The application object defines a relationship where the related View_area is derived from the relating View_area;

- 'hierarchy': The application object defines a hierarchical relationship where the related View_area is on a lower level than the relating View_area;

- 'substitution': The application object defines a relationship where the related View_area replaces the relating View_area.

4.2.494 View_placed_annotation

A View_placed_annotation is an annotation that is located in the coordinate system of a View (see 4.2.490).

The data associated with a View_placed_annotation are the following:

- appearance;

- element;
- invisible;
- reference_view.

4.2.494.1 appearance

The appearance specifies the presentation characteristics for the annotation elements of the View_placed_annotation.

See 4.3.1239 for the application assertion.

4.2.494.2 element

The element specifies the annotation element that is associated to a View_placed_annotation.

See 4.3.1238 for the application assertion.

4.2.494.3 invisible

The invisible specifies if the View_placed_annotation is invisible.

4.2.494.4 reference_view

The reference_view specifies the view that is associated to a View_placed_annotation.

See 4.3.1240 for the application assertion.

4.2.495 View_relationship

A View_relationship is a relationship between two View (see 4.2.490) objects.

The data associated with a View_relationship are the following:

- description;
- related;
- relating;
- relation_type.

4.2.495.1 description

The description specifies additional information about the View_relationship.

See 4.3.1241 for the application assertion.

The description need not be specified for a particular View_relationship.

If present, there shall be exactly one object that defines the description for a View_relationship.

4.2.495.2 related

The related specifies the second of the two View (see 4.2.490) objects related by the View_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.1242 for the application assertion.

4.2.495.3 relating

The relating specifies the first of the two View (see 4.2.490) objects related by the View_relationship.

NOTE – The semantic of this attribute is defined by the attribute relation_type.

See 4.3.1242 for the application assertion.

4.2.495.4 relation_type

The relation_type specifies the meaning of the relationship. Where applicable the following values shall be used:

- 'dependency': The related View (see 4.2.490) is dependent of the relating View. This means that the related View has the same view direction as the relating View.

4.2.496 Visual_appearance

A Visual_appearance is the appearance of a surface of a part regarding the optical impression.

NOTE – At least one of the attributes 'id' or 'name' shall be instantiated.

A Visual_appearance is a type of Surface_condition (see 4.2.437).

The data associated with a Visual_appearance are the following:

- colour_id;
- colour_name;
- id;
- lustre;
- name;
- pattern;
- transparency.

4.2.496.1 colour_id

The colour_id specifies the identifier of the colour of the Visual_appearance.

EXAMPLE 232 – A colour can be identified using a code such as 'C130202250'.

4.2.496.2 colour_name

The colour_name specifies the word or group of words by which the colour of the Visual_-appearance is referred to.

EXAMPLE 233 – A colour may be referred to, e.g., by a customer, by name such as 'arctic white'.

See 4.3.1243 for the application assertion.

The colour_name need not be specified for a particular Visual_appearance.

If present, there shall be exactly one object that defines the colour_name for a Visual_appearance.

4.2.496.3 id

The id specifies the identifier of the Visual_appearance.

The id need not be specified for a particular Visual_appearance.

4.2.496.4 lustre

The lustre specifies the kind of shining or reflection of the surface.

4.2.496.5 name

The name specifies the word or group of words by which the Visual_appearance is referred to.

See 4.3.1243 for the application assertion.

The name need not be specified for a particular Visual_appearance.

If present, there shall be exactly one object that defines the name for a Visual_appearance.

4.2.496.6 pattern

The pattern specifies the pattern used for the definition of the Visual_appearance.

EXAMPLE 234 – A pattern may be used to specify the texture of a seat cover for visual appearance purposes.

The pattern need not be specified for a particular Visual_appearance.

4.2.496.7 transparency

The transparency specifies the grade of transparency of the surface.

The transparency need not be specified for a particular Visual_appearance.

4.2.497 Wireframe_model_2d

A Wireframe_model_2d is a collection of Curve_set_2d (see 4.2.105) objects, i.e. Curve (see 4.2.102) and Point (see 4.2.322) objects defined in two-dimensional space.

A Wireframe_model_2d is a type of Geometric_model (see 4.2.211).

The data associated with a Wireframe_model_2d are the following:

- element.

The element specifies the set of Curve_set_2d (see 4.2.105) objects that form the Wireframe_model_2d.

See 4.3.1244 for the application assertion.

4.2.498 Wireframe_model_3d

A Wireframe_model_3d is a collection of Curve_set_3d (see 4.2.106) objects, i.e. Curve (see 4.2.102) and Point (see 4.2.322) objects defined in three-dimensional space.

A Wireframe_model_3d is a type of Geometric_model (see 4.2.211).

The data associated with a Wireframe_model_3d are the following:

- element.

The element specifies the set of Curve_set_3d (see 4.2.106) objects that form the Wireframe_model_3d.

See 4.3.1245 for the application assertion.

4.2.499 Woodruff_slot_end_type

A Woodruff_slot_end_type is an end condition of a slot defined by a radius tangent to the bottom of the slot and curved towards the surface of the part.

EXAMPLE 235 – Figure 65 depicts a Woodruff_slot_end_type.

A Woodruff_slot_end_type is a type of Slot_end_type (see 4.2.406).

The data associated with a Woodruff_slot_end_type are the following:

- radius.

The radius specifies the radius of the Woodruff_slot_end_type.

See 4.3.1246 for the application assertion.

4.2.500 Work_order

A Work_order is the authorization for an Activity (see 4.2.2) to be performed.

The data associated with a Work_order are the following:

- description;

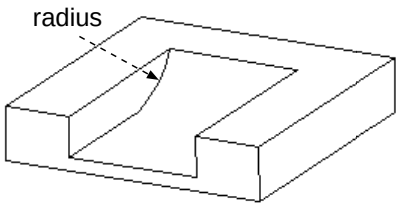


Figure 65 – Woodruff_slot_end_type

- id;
- internal;
- is_controlling;
- version_id;
- work_order_type.

4.2.500.1 description

The description specifies additional information about the Work_order (see 4.2.500).

See 4.3.1248 for the application assertion.

The description need not be specified for a particular Work_order.

If present, there shall be exactly one object that defines the description for a Work_order.

4.2.500.2 id

The id specifies the identifier of the Work_order.

4.2.500.3 internal

The internal specifies whether the Activity (see 4.2.2) objects controlled by this particular Work_order are performed internally within the organization or externally by another organization.

4.2.500.4 is_controlling

The is_controlling specifies the Activity (see 4.2.2) that is controlled by this particular Work_order.

See 4.3.1247 for the application assertion.

4.2.500.5 version_id

The version_id specifies the identification of a particular version of a Work_order.

The version_id need not be specified for a particular Work_order.

4.2.500.6 work_order_type

The work_order_type specifies the kind of the Work_order. Where applicable the following values shall be used:

- 'design release': An authorization for the design of a product or of an item and/or to create a bill of material;

- 'manufacturing release': An authorization for the manufacturing process of a product or of an item;

- 'design deviation permit': An authorization for a deviation from the approved design data;

- 'production deviation permit': An authorization for a deviation from the approved manufacturing process;

- 'management resolution': An authorization by a committee like the board of directors to design or change a product, a product_component or some items.

4.2.501 Work_request

A Work_request is the solicitation for some type of work to be done.

The data associated with a Work_request are the following:

- description;

- id;

- notified_person;

- recommended_method;

- request_type;

- requestor;

- scope;

- status;

- version_id.

4.2.501.1 description

The description specifies additional information about the Work_request.

See 4.3.1270 for the application assertion.

There shall be exactly one object that defines the description for a Work_request.

4.2.501.2 id

The id specifies the identifier of the Work_request.

4.2.501.3 notified_person

The notified_person specifies the person or the organization who shall be informed about the Work_request and the date when the person or organization was informed.

See 4.3.1257 for the application assertion.

4.2.501.4 recommended_method

The recommended_method specifies an Activity_method (see 4.2.4) that is recommended to solve the problem that the Work_request addresses.

See 4.3.1249 for the application assertion.

4.2.501.5 request_type

The request_type specifies the intention of the Work_request. Where applicable the following values shall be used:

- 'technical improvement': A request aiming at the technical improvement of an item;
- 'government regulation': A request resulting from legal requirements;
- 'durability improvement': A request aiming at a longer life time of an item;
- 'tool improvement': A request aiming at a reduction of the wear out of tools;
- 'production relief': A request aiming at a simpler assembly and production process;
- 'production requirement': A request for an activity necessary from a production point of view;
- 'quality improvement': A request aiming at an increased quality of an item;
- 'security reason': A request for an activity necessary from a security point of view;

– 'standardization': A request to unify variants of an item;

– 'cost reduction': A request aiming at the reduction of engineering and manufacturing costs of an item;

– 'customer rejection': A request resulting from a rejection by a customer;

– 'supplier request': A request resulting from requests by a supplier;

– 'customer request': A request resulting from requests by a customer;

– 'change of standard': A request to translate a standard change into action;

– 'production alignment': A request to adjust the manufacturing process of different items;

– 'procurement alignment': A request to adjust the purchasing process of different items.

4.2.501.6 requestor

The requestor specifies the person or organization who issued the Work_request and the date when this person or organization issued the Work_request.

See 4.3.1257 for the application assertion.

The requestor need not be specified for a particular Work_request.

4.2.501.7 scope

The scope specifies the objects that are affected by the Work_request.

See 4.3.1249, 4.3.1250, 4.3.1251, 4.3.1252, 4.3.1253, 4.3.1254, 4.3.1255, 4.3.1256, 4.3.1258, 4.3.1259, 4.3.1260, 4.3.1261, 4.3.1262, 4.3.1263, 4.3.1264, 4.3.1265, 4.3.1266, 4.3.1267, 4.3.1268, 4.3.1269, 4.3.1271, 4.3.1272, 4.3.1273, 4.3.1274, 4.3.1275, 4.3.1276, 4.3.1277, 4.3.1278, 4.3.1279, 4.3.1280, 4.3.1281, 4.3.1282, 4.3.1283, 4.3.1284, and 4.3.1285 for the application assertions.

4.2.501.8 status

The status specifies the status of the Work_request.

EXAMPLE 236 – 'ongoing' would be an example for status.

NOTE – The values and sequence of status are company specific.

4.2.501.9 version_id

The version_id specifies the identification of a particular version of a Work_request.

The version_id need not be specified for a particular Work_request.

4.3 Application assertions

This subclause specifies the application assertions for the Core Data for Automotive Mechanical Design Processes application protocol. Application assertions specify the relationships between application objects, the cardinality of the relationships, and the rules required for the integrity and validity of the application objects and UoFs. The application assertions and their definitions are given below.

4.3.1 Accuracy to Face

Each Accuracy is_defined_for zero or more Face objects. Each Face is related to zero or more Accuracy objects.

NOTE – This assertion is established through accuracy_select.

4.3.2 Accuracy to Geometric_model

Each Accuracy is_defined_for zero or more Geometric_model objects. Each Geometric_model is related to zero or more Accuracy objects.

NOTE – This assertion is established through accuracy_select.

4.3.3 Accuracy to Oriented_edge_curve

Each Accuracy is_defined_for zero or more Oriented_edge_curve objects. Each Oriented_edge_curve is related to zero or more Accuracy objects.

NOTE – This assertion is established through accuracy_select.

4.3.4 Accuracy to Vertex_point

Each Accuracy is_defined_for zero or more Vertex_point objects. Each Vertex_point is related to zero or more Accuracy objects.

NOTE – This assertion is established through accuracy_select.

4.3.5 Activity to Activity_method

Each Activity refers to zero or one Activity_method in the role of chosen_method. Each Activity_method acts as chosen_method for zero or more Activity objects.

4.3.6 Activity to Date_and_person_or_organization

Each Activity refers to zero or one Date_and_person_or_organization in the role of requestor. Each Date_and_person_or_organization acts as requestor for zero or more Activity objects.

4.3.7 Activity to Date_time

Each Activity refers to zero or one Date_time in the role of actual_end_date. Each Date_time acts as actual_end_date for zero or more Activity objects.

Each Activity refers to zero or one Date_time in the role of actual_start_date. Each Date_time acts as actual_start_date for zero or more Activity objects.

Each Activity refers to zero or one Date_time in the role of planned_end_date. Each Date_time acts as planned_end_date for zero or more Activity objects.

NOTE – This assertion is established through period_or_date_select.

Each Activity refers to zero or one Date_time in the role of planned_start_date. Each Date_time acts as planned_start_date for zero or more Activity objects.

NOTE – This assertion is established through event_or_date_select.

4.3.8 Activity to Event_reference

Each Activity refers to zero or one Event_reference in the role of planned_end_date. Each Event_reference acts as planned_end_date for zero or more Activity objects.

NOTE – This assertion is established through period_or_date_select.

Each Activity refers to zero or one Event_reference in the role of planned_start_date. Each Event_reference acts as planned_start_date for zero or more Activity objects.

NOTE – This assertion is established through event_or_date_select.

4.3.9 Activity to Multi_language_string

Each Activity refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Activity objects.

NOTE – This assertion is established through string_select.

4.3.10 Activity to Organization

Each Activity refers to zero or more Organization objects in the role of concerned_organization. Each Organization acts as concerned_organization for zero or more Activity objects.

Each Activity refers to zero or more Organization objects in the role of supplying_organization. Each Organization acts as supplying_organization for zero or more Activity objects.

4.3.11 Activity to Value_with_unit

Each Activity refers to zero or one Value_with_unit in the role of planned_end_date. Each Value_with_unit acts as planned_end_date for zero or more Activity objects.

NOTE – This assertion is established through period_or_date_select.

4.3.12 Activity to Work_request

Each Activity refers to zero or more Work_request objects in the role of resolved_request. Each Work_request acts as resolved_request for zero or more Activity objects.

4.3.13 Activity_element to Activity

Each Activity_element refers to exactly one Activity in the role of activity. Each Activity acts as activity for zero or more Activity_element objects.

4.3.14 Activity_element to Activity_method

Each Activity_element refers to zero or one Activity_method in the role of element. Each Activity_method acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select.

4.3.15 Activity_element to Assembly_substitute_relationship

Each Activity_element refers to zero or one Assembly_substitute_relationship in the role of element. Each Assembly_substitute_relationship acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_effectivity_select.

4.3.16 Activity_element to Class_category_association

Each Activity_element refers to zero or one Class_category_association in the role of element. Each Class_category_association acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and configuration_information_select.

4.3.17 Activity_element to Class_condition_association

Each Activity_element refers to zero or one Class_condition_association in the role of element. Each Class_condition_association acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and configuration_information_select.

4.3.18 Activity_element to Class_function_association

Each Activity_element refers to zero or one Class_function_association in the role of element. Each Class_function_association acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and specification_information_select.

4.3.19 Activity_element to Class_inclusion_association

Each Activity_element refers to zero or one Class_inclusion_association in the role of element. Each Class_inclusion_association acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and configuration_information_select.

4.3.20 Activity_element to Class_specification_association

Each Activity_element refers to zero or one Class_specification_association in the role of element. Each Class_specification_association acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and configuration_information_select.

4.3.21 Activity_element to Configuration

Each Activity_element refers to zero or one Configuration in the role of element. Each Configuration acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select.

4.3.22 Activity_element to Design_constraint

Each Activity_element refers to zero or one Design_constraint in the role of element. Each Design_constraint acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and specification_information_select.

4.3.23 Activity_element to Drawing

Each Activity_element refers to zero or one Drawing in the role of element. Each Drawing acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_effectivity_select.

4.3.24 Activity_element to External_document

Each Activity_element refers to zero or one External_document in the role of element. Each External_document acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_effectivity_select.

4.3.25 Activity_element to Geometric_model

Each Activity_element refers to zero or one Geometric_model in the role of element. Each Geometric_model acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and model_select.

4.3.26 Activity_element to Item

Each Activity_element refers to zero or one Item in the role of element. Each Item acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_effectivity_select.

4.3.27 Activity_element to Item_definition_instance_relationship

Each Activity_element refers to zero or one Item_definition_instance_relationship in the role of element. Each Item_definition_instance_relationship acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and item_relationship_select.

4.3.28 Activity_element to Item_definition_relationship

Each Activity_element refers to zero or one Item_definition_relationship in the role of element. Each Item_definition_relationship acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and item_relationship_select.

4.3.29 Activity_element to Item_instance

Each Activity_element refers to zero or one Item_instance in the role of element. Each Item_instance acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_effectivity_select.

4.3.30 Activity_element to Item_instance_relationship

Each Activity_element refers to zero or one Item_instance_relationship in the role of element. Each Item_instance_relationship acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and item_relationship_select.

4.3.31 Activity_element to Item_solution

Each Activity_element refers to zero or one Item_solution in the role of element. Each Item_solution acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and specification_information_select.

4.3.32 Activity_element to Item_version

Each Activity_element refers to zero or one Item_version in the role of element. Each Item_version acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_effectivity_select.

4.3.33 Activity_element to Material

Each Activity_element refers to zero or one Material in the role of element. Each Material acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_effectivity_select.

4.3.34 Activity_element to Physical_instance

Each Activity_element refers to zero or one Physical_instance in the role of element. Each Physical_instance acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select.

4.3.35 Activity_element to Physical_instance_usage

Each Activity_element refers to zero or one Physical_instance_usage in the role of element. Each Physical_instance_usage acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select.

4.3.36 Activity_element to Process_plan

Each Activity_element refers to zero or one Process_plan in the role of element. Each Process_plan acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_effectivity_select.

4.3.37 Activity_element to Product_class

Each Activity_element refers to zero or one Product_class in the role of element. Each Product_class acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.38 Activity_element to Product_component

Each Activity_element refers to zero or one Product_component in the role of element. Each Product_component acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.39 Activity_element to Product_function

Each Activity_element refers to zero or one Product_function in the role of element. Each Product_function acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.40 Activity_element to Product_specification

Each Activity_element refers to zero or one Product_specification in the role of element. Each Product_specification acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_-effectivity_select.

4.3.41 Activity_element to Property

Each Activity_element refers to zero or one Property in the role of element. Each Property acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_-effectivity_select.

4.3.42 Activity_element to Simulation_information

Each Activity_element refers to zero or one Simulation_information in the role of element. Each Simulation_information acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and model_select.

4.3.43 Activity_element to Specification

Each Activity_element refers to zero or one Specification in the role of element. Each Specification acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.44 Activity_element to Specification_category

Each Activity_element refers to zero or one Specification_category in the role of element. Each Specification_category acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and specification_information_select.

4.3.45 Activity_element to Specification_expression

Each Activity_element refers to zero or one Specification_expression in the role of element. Each Specification_expression acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and configuration_information_select.

4.3.46 Activity_element to Specification_inclusion

Each Activity_element refers to zero or one Specification_inclusion in the role of element. Each Specification_inclusion acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and configuration_information_select.

4.3.47 Activity_element to Styled_model

Each Activity_element refers to zero or one Styled_model in the role of element. Each Styled_model acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and model_select.

4.3.48 Activity_element to View_area

Each Activity_element refers to zero or one View_area in the role of element. Each View_area acts as element for zero or more Activity_element objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_effectivity_select.

4.3.49 Activity_method to Multi_language_string

Each Activity_method refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Activity_method objects.

NOTE – This assertion is established through string_select.

Each Activity_method refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Activity_method objects.

NOTE – This assertion is established through string_select.

4.3.50 Activity_relationship to Activity

Each Activity_relationship is related to exactly one Activity. Each Activity is referenced by zero or more Activity_relationship objects as related.

NOTE – The related Activity is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Activity_relationship is relating to exactly one Activity. Each Activity is referenced by zero or more Activity_relationship objects as relating.

NOTE – The relating Activity is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.51 Activity_relationship to Multi_language_string

Each Activity_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Activity_relationship objects.

NOTE – This assertion is established through string_select.

4.3.52 Actuator to Kinematic_pair

Each Actuator is_applied_to exactly one Kinematic_pair. Each Kinematic_pair is_referenced_by zero or one Actuator.

4.3.53 Actuator to Multi_language_string

Each Actuator refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Actuator objects.

NOTE – This assertion is established through string_select.

4.3.54 Alias_identification to Approval_status

Each Alias_identification is_applied_to zero or one Approval_status. Each Approval_status is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.55 Alias_identification to Classification_system

Each Alias_identification is_applied_to zero or one Classification_system. Each Classification_system is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.56 Alias_identification to Document_type

Each Alias_identification is_applied_to zero or one Document_type. Each Document_type is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.57 Alias_identification to External_document

Each Alias_identification is_applied_to zero or one External_document. Each External_document is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.58 Alias_identification to Geometric_model

Each Alias_identification is_applied_to zero or one Geometric_model. Each Geometric_model is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.59 Alias_identification to Item_instance

Each Alias_identification is_applied_to zero or one Item_instance. Each Item_instance is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.60 Alias_identification to Item_solution

Each Alias_identification is_applied_to zero or one Item_solution. Each Item_solution is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.61 Alias_identification to Multi_language_string

Each Alias_identification refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Alias_identification objects.

NOTE – This assertion is established through string_select.

4.3.62 Alias_identification to Organization

Each Alias_identification refers to zero or one Organization in the role of alias_scope. Each Organization acts as alias_scope for zero or more Alias_identification objects.

4.3.63 Alias_identification to Physical_instance

Each Alias_identification is_applied_to zero or one Physical_instance. Each Physical_instance is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.64 Alias_identification to Product_class

Each Alias_identification is_applied_to zero or one Product_class. Each Product_class is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.65 Alias_identification to Product_component

Each Alias_identification is_applied_to zero or one Product_component. Each Product_component is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.66 Alias_identification to Product_function

Each Alias_identification is_applied_to zero or one Product_function. Each Product_function is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.67 Alias_identification to Security_level

Each Alias_identification is_applied_to zero or one Security_level. Each Security_level is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.68 Alias_identification to Specification

Each Alias_identification is_applied_to zero or one Specification. Each Specification is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.69 Alias_identification to Specification_category

Each Alias_identification is_applied_to zero or one Specification_category. Each Specification_category is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.70 Alias_identification to Styled_model

Each Alias_identification is_applied_to zero or one Styled_model. Each Styled_model is related to zero or more Alias_identification objects.

NOTE – This assertion is established through alias_select.

4.3.71 Alternate_item_relationship to Item

Each Alternate_item_relationship refers to exactly one Item in the role of alternate. Each Item acts as alternate for zero or more Alternate_item_relationship objects.

Each Alternate_item_relationship refers to exactly one Item in the role of base. Each Item acts as base for zero or more Alternate_item_relationship objects.

4.3.72 Alternate_item_relationship to Multi_language_string

Each Alternate_item_relationship refers to zero or one Multi_language_string in the role of fulfilled_requirements. Each Multi_language_string acts as fulfilled_requirements for zero or more Alternate_item_relationship objects.

NOTE – This assertion is established through string_select.

4.3.73 Angle_specification to Feature_parameter

Each Angle_specification refers to exactly one Feature_parameter in the role of angle. Each Feature_parameter acts as angle for zero or more Angle_specification objects.

4.3.74 Angle_taper to Feature_parameter

Each Angle_taper refers to exactly one Feature_parameter in the role of angle. Each Feature_parameter acts as angle for zero or more Angle_taper objects.

4.3.75 Angularity_tolerance to Datum

Each Angularity_tolerance refers to exactly one Datum in the role of datum. Each Datum acts as datum for zero or more Angularity_tolerance objects.

4.3.76 Annotation_model to Cartesian_coordinate_space

Each Annotation_model is_defined_in exactly one Cartesian_coordinate_space. Each Cartesian_coordinate_space is related to zero or more Annotation_model objects.

4.3.77 Appearance_assignment to Curve_appearance

Each Appearance_assignment refers to zero or more Curve_appearance objects in the role of component. Each Curve_appearance acts as component for one or more Appearance_assignment objects.

NOTE – This assertion is established through appearance_select.

4.3.78 Appearance_assignment to Hatching

Each Appearance_assignment refers to zero or more Hatching objects in the role of component. Each Hatching acts as component for one or more Appearance_assignment objects.

NOTE – This assertion is established through appearance_select.

4.3.79 Appearance_assignment to Point_appearance

Each Appearance_assignment refers to zero or more Point_appearance objects in the role of component. Each Point_appearance acts as component for one or more Appearance_assignment objects.

NOTE – This assertion is established through appearance_select.

4.3.80 Appearance_assignment to Solid_fill_area

Each Appearance_assignment refers to zero or more Solid_fill_area objects in the role of component. Each Solid_fill_area acts as component for one or more Appearance_assignment objects.

NOTE – This assertion is established through appearance_select.

4.3.81 Appearance_assignment to Surface_appearance

Each Appearance_assignment refers to zero or more Surface_appearance objects in the role of component. Each Surface_appearance acts as component for one or more Appearance_assignment objects.

NOTE – This assertion is established through appearance_select.

4.3.82 Appearance_assignment to Text_appearance

Each Appearance_assignment refers to zero or more Text_appearance objects in the role of component. Each Text_appearance acts as component for one or more Appearance_assignment objects.

NOTE – This assertion is established through appearance_select.

4.3.83 Appearance_assignment to Tiling

Each Appearance_assignment refers to zero or more Tiling objects in the role of component. Each Tiling acts as component for one or more Appearance_assignment objects.

NOTE – This assertion is established through appearance_select.

4.3.84 Appearance_assignment to Vector_appearance

Each Appearance_assignment refers to zero or more Vector_appearance objects in the role of component. Each Vector_appearance acts as component for one or more Appearance_assignment objects.

NOTE – This assertion is established through appearance_select.

4.3.85 Application_context to Multi_language_string

Each Application_context refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Application_context objects.

NOTE – This assertion is established through string_select.

4.3.86 Approval to Activity

Each Approval is_applied_to zero or more Activity objects. Each Activity is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.87 Approval to Activity_element

Each Approval is_applied_to zero or more Activity_element objects. Each Activity_element is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.88 Approval to Alternate_item_relationship

Each Approval is_applied_to zero or more Alternate_item_relationship objects. Each Alternate_item_relationship is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.89 Approval to Approval_status

Each Approval refers to exactly one Approval_status in the role of approval_status. Each Approval_status acts as approval_status for zero or more Approval objects.

4.3.90 Approval to Assembly_substitute_relationship

Each Approval is_applied_to zero or more Assembly_substitute_relationship objects. Each Assembly_substitute_relationship is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.91 Approval to Class_category_association

Each Approval is_applied_to zero or more Class_category_association objects. Each Class_category_association is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and configuration_information_select.

4.3.92 Approval to Class_condition_association

Each Approval is_applied_to zero or more Class_condition_association objects. Each Class_condition_association is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and configuration_information_select.

4.3.93 Approval to Class_function_association

Each Approval is_applied_to zero or more Class_function_association objects. Each Class_function_association is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and specification_information_select.

4.3.94 Approval to Class_inclusion_association

Each Approval is_applied_to zero or more Class_inclusion_association objects. Each Class_inclusion_association is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and configuration_information_select.

4.3.95 Approval to Class_specification_association

Each Approval is_applied_to zero or more Class_specification_association objects. Each Class_specification_association is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and configuration_information_select.

4.3.96 Approval to Configuration

Each Approval is_applied_to zero or more Configuration objects. Each Configuration is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.97 Approval to Date_and_person_or_organization

Each Approval is_approved_by zero or more Date_and_person_or_organization objects. Each Date_and_person_or_organization is related to zero or more Approval objects.

4.3.98 Approval to Date_time

Each Approval refers to zero or one Date_time in the role of actual_date. Each Date_time acts as actual_date for zero or more Approval objects.

Each Approval refers to zero or one Date_time in the role of nominal_date. Each Date_time acts as nominal_date for zero or more Approval objects.

4.3.99 Approval to Design_constraint

Each Approval is_applied_to zero or more Design_constraint objects. Each Design_constraint is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and specification_information_select.

4.3.100 Approval to Design_discipline_item_definition

Each Approval is_applied_to zero or more Design_discipline_item_definition objects. Each Design_discipline_item_definition is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and item_information_select.

4.3.101 Approval to Drawing

Each Approval is_applied_to zero or more Drawing objects. Each Drawing is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.102 Approval to External_document

Each Approval is_applied_to zero or more External_document objects. Each External_document is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.103 Approval to Geometric_model

Each Approval is_applied_to zero or more Geometric_model objects. Each Geometric_model is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and model_select.

4.3.104 Approval to Item_definition_instance_relationship

Each Approval is_applied_to zero or more Item_definition_instance_relationship objects. Each Item_definition_instance_relationship is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select, item_information_select, and item_relationship_select.

4.3.105 Approval to Item_definition_relationship

Each Approval is_applied_to zero or more Item_definition_relationship objects. Each Item_definition_relationship is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select, item_information_select, and item_relationship_select.

4.3.106 Approval to Item_instance

Each Approval is_applied_to zero or more Item_instance objects. Each Item_instance is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and item_information_select.

4.3.107 Approval to Item_instance_relationship

Each Approval is_applied_to zero or more Item_instance_relationship objects. Each Item_instance_relationship is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select, item_information_select, and item_relationship_select.

4.3.108 Approval to Item_solution

Each Approval is_applied_to zero or more Item_solution objects. Each Item_solution is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and specification_information_select.

4.3.109 Approval to Item_version

Each Approval is_applied_to zero or more Item_version objects. Each Item_version is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.110 Approval to Material

Each Approval is_applied_to zero or more Material objects. Each Material is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.111 Approval to Multi_language_string

Each Approval refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Approval objects.

NOTE – This assertion is established through string_select.

4.3.112 Approval to Physical_instance

Each Approval is_applied_to zero or more Physical_instance objects. Each Physical_instance is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.113 Approval to Physical_instance_usage

Each Approval is_applied_to zero or more Physical_instance_usage objects. Each Physical_instance_usage is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.114 Approval to Process_plan

Each Approval is_applied_to zero or more Process_plan objects. Each Process_plan is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.115 Approval to Product_class

Each Approval is_applied_to zero or more Product_class objects. Each Product_class is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and specification_information_select.

4.3.116 Approval to Product_component

Each Approval is_applied_to zero or more Product_component objects. Each Product_component is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and specification_information_-select.

4.3.117 Approval to Product_function

Each Approval is_applied_to zero or more Product_function objects. Each Product_function is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and specification_information_-select.

4.3.118 Approval to Project

Each Approval is_applied_to zero or more Project objects. Each Project is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.119 Approval to Simulation_information

Each Approval is_applied_to zero or more Simulation_information objects. Each Simulation_-information is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and model_select.

4.3.120 Approval to Specification

Each Approval is_applied_to zero or more Specification objects. Each Specification is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and specification_information_-select.

4.3.121 Approval to Specification_category

Each Approval is_applied_to zero or more Specification_category objects. Each Specification_-category is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and specification_information_-select.

4.3.122 Approval to Specification_expression

Each Approval is_applied_to zero or more Specification_expression objects. Each Specification_-expression is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and configuration_information_-select.

4.3.123 Approval to Specification_inclusion

Each Approval is_applied_to zero or more Specification_inclusion objects. Each Specification_inclusion is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and configuration_information_select.

4.3.124 Approval to Styled_model

Each Approval is_applied_to zero or more Styled_model objects. Each Styled_model is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select and model_select.

4.3.125 Approval to View_area

Each Approval is_applied_to zero or more View_area objects. Each View_area is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.126 Approval to Work_order

Each Approval is_applied_to zero or more Work_order objects. Each Work_order is related to zero or more Approval objects.

NOTE – This assertion is established through approval_element_select.

4.3.127 Approval_relationship to Approval

Each Approval_relationship is related to exactly one Approval. Each Approval is referenced by zero or more Approval_relationship objects as related.

NOTE – The related Approval is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Approval_relationship is relating to exactly one Approval. Each Approval is referenced by zero or more Approval_relationship objects as relating.

NOTE – The relating Approval is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.128 Approval_status to Classification_system

Each Approval_status refers to zero or one Classification_system in the role of used_classification_system. Each Classification_system acts as used_classification_system for zero or more Approval_status objects.

4.3.129 Area_placed_annotation to Annotation_element

Each Area_placed_annotation refers to exactly one Annotation_element in the role of element. Each Annotation_element acts as element for zero or more Area_placed_annotation objects.

4.3.130 Area_placed_annotation to Appearance_assignment

Each Area_placed_annotation refers to exactly one Appearance_assignment in the role of appearance. Each Appearance_assignment acts as appearance for zero or more Area_placed_annotation objects.

4.3.131 Area_placed_annotation to Drawing_sheet

Each Area_placed_annotation refers to zero or one Drawing_sheet in the role of reference_area. Each Drawing_sheet acts as reference_area for zero or more Area_placed_annotation objects.

NOTE – This assertion is established through area_select.

4.3.132 Area_placed_annotation to View_area

Each Area_placed_annotation refers to zero or one View_area in the role of reference_area. Each View_area acts as reference_area for zero or more Area_placed_annotation objects.

NOTE – This assertion is established through area_select.

4.3.133 Assembly_component_relationship to Assembly_definition

Each Assembly_component_relationship is relating to exactly one Assembly_definition. Each Assembly_definition is referenced by zero or more Assembly_component_relationship objects as relating.

NOTE – The relating Assembly_definition is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.134 Assembly_component_relationship to Geometric_model_relationship_with_transformation

Each Assembly_component_relationship refers to zero or one Geometric_model_relationship_with_transformation in the role of transformation. Each Geometric_model_relationship_with_transformation acts as transformation for zero or more Assembly_component_relationship objects.

NOTE – This assertion is established through transformation_select.

4.3.135 Assembly_component_relationship to Template_instance

Each Assembly_component_relationship refers to zero or one Template_instance in the role of transformation. Each Template_instance acts as transformation for zero or more Assembly_component_relationship objects.

NOTE – This assertion is established through transformation_select.

4.3.136 Assembly_substitute_relationship to Assembly_component_relationship

Each Assembly_substitute_relationship refers to exactly one Assembly_component_relationship in the role of base. Each Assembly_component_relationship acts as base for zero or more Assembly_substitute_relationship objects.

Each Assembly_substitute_relationship refers to exactly one Assembly_component_relationship in the role of substitute. Each Assembly_component_relationship acts as substitute for zero or more Assembly_substitute_relationship objects.

4.3.137 Assembly_substitute_relationship to Multi_language_string

Each Assembly_substitute_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Assembly_substitute_relationship objects.

NOTE – This assertion is established through string_select.

4.3.138 Associated_annotation_symbol to Annotation_curve

Each Associated_annotation_symbol is_associated_to zero or more Annotation_curve objects. Each Annotation_curve is related to zero or more Associated_annotation_symbol objects.

NOTE – This assertion is established through associated_element_select and dimension_associated_element_select.

4.3.139 Associated_annotation_symbol to Annotation_subfigure

Each Associated_annotation_symbol is_associated_to zero or more Annotation_subfigure objects. Each Annotation_subfigure is related to zero or more Associated_annotation_symbol objects.

NOTE – This assertion is established through associated_element_select.

4.3.140 Associated_annotation_symbol to Annotation_symbol

Each Associated_annotation_symbol is_associated_to zero or more Annotation_symbol objects. Each Annotation_symbol is related to zero or more Associated_annotation_symbol objects.

NOTE – This assertion is established through associated_element_select.

4.3.141 Associated_annotation_symbol to Detailed_geometric_model_element

Each Associated_annotation_symbol is_associated_to zero or more Detailed_geometric_model_element objects. Each Detailed_geometric_model_element is related to zero or more Associated_annotation_symbol objects.

NOTE – This assertion is established through associated_element_select and dimension_associated_element_select.

4.3.142 Associated_annotation_symbol to Fill_area

Each Associated_annotation_symbol is_associated_to zero or more Fill_area objects. Each Fill_area is related to zero or more Associated_annotation_symbol objects.

NOTE – This assertion is established through associated_element_select.

4.3.143 Associated_annotation_symbol to Text

Each Associated_annotation_symbol is_associated_to zero or more Text objects. Each Text is related to zero or more Associated_annotation_symbol objects.

NOTE – This assertion is established through associated_element_select.

4.3.144 Associated_fill_area to Annotation_curve

Each Associated_fill_area is_derived_from zero or more Annotation_curve objects. Each Annotation_curve is related to zero or more Associated_fill_area objects.

NOTE – This assertion is established through dimension_associated_element_select.

4.3.145 Associated_fill_area to Detailed_geometric_model_element

Each Associated_fill_area is_derived_from zero or more Detailed_geometric_model_element objects. Each Detailed_geometric_model_element is related to zero or more Associated_fill_area objects.

NOTE – This assertion is established through dimension_associated_element_select.

4.3.146 Associated_leader to Annotation_curve

Each Associated_leader refers to zero or one Annotation_curve in the role of target. Each Annotation_curve acts as target for zero or more Associated_leader objects.

NOTE – This assertion is established through associated_element_select and dimension_associated_element_select.

4.3.147 Associated_leader to Annotation_subfigure

Each Associated_leader refers to zero or one Annotation_subfigure in the role of target. Each Annotation_subfigure acts as target for zero or more Associated_leader objects.

NOTE – This assertion is established through associated_element_select.

4.3.148 Associated_leader to Annotation_symbol

Each Associated_leader refers to zero or one Annotation_symbol in the role of target. Each Annotation_symbol acts as target for zero or more Associated_leader objects.

NOTE – This assertion is established through associated_element_select.

4.3.149 Associated_leader to Detailed_geometric_model_element

Each Associated_leader refers to zero or one Detailed_geometric_model_element in the role of target. Each Detailed_geometric_model_element acts as target for zero or more Associated_-leader objects.

NOTE – This assertion is established through associated_element_select and dimension_associated_-element_select.

4.3.150 Associated_leader to Fill_area

Each Associated_leader refers to zero or one Fill_area in the role of target. Each Fill_area acts as target for zero or more Associated_leader objects.

NOTE – This assertion is established through associated_element_select.

4.3.151 Associated_leader to Text

Each Associated_leader refers to zero or one Text in the role of target. Each Text acts as target for zero or more Associated_leader objects.

NOTE – This assertion is established through associated_element_select.

4.3.152 Associated_projection_curve to Annotation_curve

Each Associated_projection_curve is_projected_to zero or one Annotation_curve. Each Annotation_curve is related to zero or more Associated_projection_curve objects.

NOTE – This assertion is established through associated_element_select and dimension_associated_-element_select.

4.3.153 Associated_projection_curve to Annotation_subfigure

Each Associated_projection_curve is_projected_to zero or one Annotation_subfigure. Each Annotation_subfigure is related to zero or more Associated_projection_curve objects.

NOTE – This assertion is established through associated_element_select.

4.3.154 Associated_projection_curve to Annotation_symbol

Each Associated_projection_curve is_projected_to zero or one Annotation_symbol. Each Annotation_symbol is related to zero or more Associated_projection_curve objects.

NOTE – This assertion is established through associated_element_select.

4.3.155 Associated_projection_curve to Detailed_geometric_model_element

Each Associated_projection_curve is_projected_to zero or one Detailed_geometric_model_element. Each Detailed_geometric_model_element is related to zero or more Associated_projection_curve objects.

NOTE – This assertion is established through associated_element_select and dimension_associated_element_select.

4.3.156 Associated_projection_curve to Fill_area

Each Associated_projection_curve is_projected_to zero or one Fill_area. Each Fill_area is related to zero or more Associated_projection_curve objects.

NOTE – This assertion is established through associated_element_select.

4.3.157 Associated_projection_curve to Text

Each Associated_projection_curve is_projected_to zero or one Text. Each Text is related to zero or more Associated_projection_curve objects.

NOTE – This assertion is established through associated_element_select.

4.3.158 B_rep to Closed_shell

Each B_rep refers to exactly one Closed_shell in the role of outer. Each Closed_shell acts as outer for zero or more B_rep objects.

Each B_rep refers to zero or more Closed_shell objects in the role of void. Each Closed_shell acts as void for zero or more B_rep objects.

4.3.159 B_rep_model to B_rep

Each B_rep_model refers to zero or more B_rep objects in the role of element. Each B_rep acts as element for zero or more B_rep_model objects.

4.3.160 Bead_feature to Bead_end_type

Each Bead_feature refers to exactly two Bead_end_type objects in the role of end_condition. Each Bead_end_type acts as end_condition for zero or more Bead_feature objects.

4.3.161 Bead_feature to Feature_parameter

Each Bead_feature refers to zero or one Feature_parameter in the role of base_radius. Each Feature_parameter acts as base_radius for zero or more Bead_feature objects.

4.3.162 Bead_feature to Open_profile

Each Bead_feature refers to exactly one Open_profile in the role of bead_profile. Each Open_profile acts as bead_profile for zero or more Bead_feature objects.

4.3.163 Bead_feature to Path

Each Bead_feature refers to exactly one Path in the role of sweep_path. Each Path acts as sweep_path for zero or more Bead_feature objects.

4.3.164 Bead_feature to Transformation

Each Bead_feature refers to zero or one Transformation in the role of profile_placement. Each Transformation acts as profile_placement for zero or more Bead_feature objects.

4.3.165 Boolean_result to B_rep_model

Each Boolean_result refers to zero or one B_rep_model in the role of first_operand. Each B_rep_model acts as first_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select, csg_select, and non_primitive_solid_select.

Each Boolean_result refers to zero or one B_rep_model in the role of second_operand. Each B_rep_model acts as second_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select, csg_select, and non_primitive_solid_select.

4.3.166 Boolean_result to Boolean_result

Each Boolean_result refers to zero or one Boolean_result in the role of first_operand. Each Boolean_result acts as first_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select and csg_select.

Each Boolean_result refers to zero or one Boolean_result in the role of second_operand. Each Boolean_result acts as second_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select and csg_select.

4.3.167 Boolean_result to Csg_primitive

Each Boolean_result refers to zero or one Csg_primitive in the role of first_operand. Each Csg_primitive acts as first_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select and csg_select.

Each Boolean_result refers to zero or one Csg_primitive in the role of second_operand. Each Csg_primitive acts as second_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select and csg_select.

4.3.168 Boolean_result to Faceted_b_rep_model

Each Boolean_result refers to zero or one Faceted_b_rep_model in the role of first_operand. Each Faceted_b_rep_model acts as first_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select, csg_select, and non_primitive_solid_select.

Each Boolean_result refers to zero or one Faceted_b_rep_model in the role of second_operand. Each Faceted_b_rep_model acts as second_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select, csg_select, and non_primitive_solid_select.

4.3.169 Boolean_result to Half_space_solid

Each Boolean_result refers to zero or one Half_space_solid in the role of first_operand. Each Half_space_solid acts as first_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select.

Each Boolean_result refers to zero or one Half_space_solid in the role of second_operand. Each Half_space_solid acts as second_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select.

4.3.170 Boolean_result to Solid_replica

Each Boolean_result refers to zero or one Solid_replica in the role of first_operand. Each Solid_replica acts as first_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select, csg_select, and non_primitive_solid_select.

Each Boolean_result refers to zero or one Solid_replica in the role of second_operand. Each Solid_replica acts as second_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select, csg_select, and non_primitive_solid_select.

4.3.171 Boolean_result to Swept_face_solid

Each Boolean_result refers to zero or one Swept_face_solid in the role of first_operand. Each Swept_face_solid acts as first_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select, csg_select, and non_primitive_solid_select.

Each Boolean_result refers to zero or one Swept_face_solid in the role of second_operand. Each Swept_face_solid acts as second_operand for zero or more Boolean_result objects.

NOTE – This assertion is established through boolean_operand_select, csg_select, and non_primitive_solid_select.

4.3.172 Boss_feature to Angle_taper

Each Boss_feature refers to zero or one Angle_taper in the role of taper. Each Angle_taper acts as taper for zero or more Boss_feature objects.

NOTE – This assertion is established through taper_select.

4.3.173 Boss_feature to Boss_top_condition

Each Boss_feature refers to exactly one Boss_top_condition in the role of top_condition. Each Boss_top_condition acts as top_condition for zero or more Boss_feature objects.

4.3.174 Boss_feature to Diameter_taper

Each Boss_feature refers to zero or one Diameter_taper in the role of taper. Each Diameter_taper acts as taper for zero or more Boss_feature objects.

NOTE – This assertion is established through taper_select.

4.3.175 Boss_feature to Directed_taper

Each Boss_feature refers to zero or one Directed_taper in the role of taper. Each Directed_taper acts as taper for zero or more Boss_feature objects.

NOTE – This assertion is established through taper_select.

4.3.176 Boss_feature to Feature_parameter

Each Boss_feature refers to zero or one Feature_parameter in the role of fillet_radius. Each Feature_parameter acts as fillet_radius for zero or more Boss_feature objects.

4.3.177 Boss_feature to Linear_path

Each Boss_feature refers to exactly one Linear_path in the role of sweep_path. Each Linear_path acts as sweep_path for zero or more Boss_feature objects.

4.3.178 Boss_feature to Transformation

Each Boss_feature refers to zero or one Transformation in the role of profile_placement. Each Transformation acts as profile_placement for zero or more Boss_feature objects.

4.3.179 Catalog_thread to External_document

Each Catalog_thread is_specified_in zero or one External_document. Each External_document is related to zero or more Catalog_thread objects.

NOTE – This assertion is established through external_source_select.

4.3.180 Catalog_thread to External_library_reference

Each Catalog_thread is_specified_in zero or one External_library_reference. Each External_library_reference is related to zero or more Catalog_thread objects.

NOTE – This assertion is established through external_source_select.

4.3.181 Centre_of_mass to Cartesian_coordinate_space

Each Centre_of_mass is_defined_in exactly one Cartesian_coordinate_space. Each Cartesian_coordinate_space is related to zero or more Centre_of_mass objects.

4.3.182 Centre_of_mass to Multi_language_string

Each Centre_of_mass refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Centre_of_mass objects.

NOTE – This assertion is established through string_select.

4.3.183 Chamfer to Chamfer_definition

Each Chamfer is_defined_by exactly one Chamfer_definition. Each Chamfer_definition is related to zero or more Chamfer objects.

4.3.184 Chamfer_angle to Feature_parameter

Each Chamfer_angle refers to exactly one Feature_parameter in the role of angle. Each Feature_parameter acts as angle for zero or more Chamfer_angle objects.

4.3.185 Chamfer_definition to Chamfer_angle

Each Chamfer_definition refers to zero or one Chamfer_angle in the role of value_2. Each Chamfer_angle acts as value_2 for zero or more Chamfer_definition objects.

NOTE – This assertion is established through chamfer_value_select.

4.3.186 Chamfer_definition to Chamfer_offset

Each Chamfer_definition refers to exactly one Chamfer_offset in the role of offset_1. Each Chamfer_offset acts as offset_1 for zero or more Chamfer_definition objects.

Each Chamfer_definition refers to zero or one Chamfer_offset in the role of value_2. Each Chamfer_offset acts as value_2 for zero or more Chamfer_definition objects.

NOTE – This assertion is established through chamfer_value_select.

4.3.187 Chamfer_offset to Feature_parameter

Each Chamfer_offset refers to exactly one Feature_parameter in the role of offset. Each Feature_parameter acts as offset for zero or more Chamfer_offset objects.

4.3.188 Change to Design_constraint_relationship

Each Change refers to zero or more Design_constraint_relationship objects in the role of described_change. Each Design_constraint_relationship acts as described_change for zero or more Change objects.

NOTE – This assertion is established through change_relationship_select.

4.3.189 Change to Item_version_relationship

Each Change refers to zero or more Item_version_relationship objects in the role of described_change. Each Item_version_relationship acts as described_change for zero or more Change objects.

NOTE – This assertion is established through change_relationship_select.

4.3.190 Change to Multi_language_string

Each Change refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Change objects.

NOTE – This assertion is established through string_select.

4.3.191 Change to Process_operation_relationship

Each Change refers to zero or more Process_operation_relationship objects in the role of described_change. Each Process_operation_relationship acts as described_change for zero or more Change objects.

NOTE – This assertion is established through change_relationship_select.

4.3.192 Change to Process_plan_relationship

Each Change refers to zero or more Process_plan_relationship objects in the role of described_change. Each Process_plan_relationship acts as described_change for zero or more Change objects.

NOTE – This assertion is established through change_relationship_select.

4.3.193 Change to Product_component_relationship

Each Change refers to zero or more Product_component_relationship objects in the role of described_change. Each Product_component_relationship acts as described_change for zero or more Change objects.

NOTE – This assertion is established through change_relationship_select.

4.3.194 Change to Product_function_relationship

Each Change refers to zero or more Product_function_relationship objects in the role of described_change. Each Product_function_relationship acts as described_change for zero or more Change objects.

NOTE – This assertion is established through change_relationship_select.

4.3.195 Change to Replaced_definition_relationship

Each Change refers to zero or more Replaced_definition_relationship objects in the role of described_change. Each Replaced_definition_relationship acts as described_change for zero or more Change objects.

NOTE – This assertion is established through change_relationship_select.

4.3.196 Change to Replaced_usage_relationship

Each Change refers to zero or more Replaced_usage_relationship objects in the role of described_change. Each Replaced_usage_relationship acts as described_change for zero or more Change objects.

NOTE – This assertion is established through change_relationship_select.

4.3.197 Change to Shape_aspect_relationship

Each Change refers to zero or more Shape_aspect_relationship objects in the role of described_change. Each Shape_aspect_relationship acts as described_change for zero or more Change objects.

NOTE – This assertion is established through change_relationship_select.

4.3.198 Circular_boss to Circular_closed_profile

Each Circular_boss refers to exactly one Circular_closed_profile in the role of boundary. Each Circular_closed_profile acts as boundary for zero or more Circular_boss objects.

4.3.199 Circular_closed_profile to Feature_parameter

Each Circular_closed_profile refers to exactly one Feature_parameter in the role of diameter. Each Feature_parameter acts as diameter for zero or more Circular_closed_profile objects.

4.3.200 Circular_path to Axis_placement

Each Circular_path refers to exactly one Axis_placement in the role of rotation_axis_defined_by. Each Axis_placement acts as rotation_axis_defined_by for zero or more Circular_path objects.

4.3.201 Circular_path to Feature_parameter

Each Circular_path refers to exactly one Feature_parameter in the role of radius. Each Feature_parameter acts as radius for zero or more Circular_path objects.

4.3.202 Circular_pattern to Circular_offset_pattern

Each Circular_pattern refers to zero or more Circular_offset_pattern objects in the role of relocated_feature. Each Circular_offset_pattern acts as relocated_feature for zero or more Circular_pattern objects.

4.3.203 Circular_pattern to Circular_omit_pattern

Each Circular_pattern refers to zero or more Circular_omit_pattern objects in the role of missing_feature. Each Circular_omit_pattern acts as missing_feature for zero or more Circular_pattern objects.

4.3.204 Circular_pattern to Feature_parameter

Each Circular_pattern refers to exactly one Feature_parameter in the role of angular_spacing. Each Feature_parameter acts as angular_spacing for zero or more Circular_pattern objects.

Each Circular_pattern refers to zero or one Feature_parameter in the role of diameter. Each Feature_parameter acts as diameter for zero or more Circular_pattern objects.

Each Circular_pattern refers to exactly one Feature_parameter in the role of rotation. Each Feature_parameter acts as rotation for zero or more Circular_pattern objects.

4.3.205 Circular_runout_tolerance to Datum

Each Circular_runout_tolerance refers to one or two Datum objects in the role of datum. Each Datum acts as datum for zero or more Circular_runout_tolerance objects.

4.3.206 Class_category_association to Product_class

Each Class_category_association refers to exactly one Product_class in the role of product_class. Each Product_class acts as product_class for zero or more Class_category_association objects.

4.3.207 Class_category_association to Specification_category

Each Class_category_association refers to exactly one Specification_category in the role of category. Each Specification_category acts as category for zero or more Class_category_association objects.

4.3.208 Class_condition_association to Multi_language_string

Each Class_condition_association refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Class_condition_association objects.

NOTE – This assertion is established through string_select.

4.3.209 Class_condition_association to Product_class

Each Class_condition_association refers to exactly one Product_class in the role of product_class. Each Product_class acts as product_class for zero or more Class_condition_association objects.

4.3.210 Class_condition_association to Specification_expression

Each Class_condition_association refers to exactly one Specification_expression in the role of condition. Each Specification_expression acts as condition for zero or more Class_condition_association objects.

4.3.211 Class_function_association to Multi_language_string

Each Class_function_association refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Class_function_association

objects.

NOTE – This assertion is established through string_select.

4.3.212 Class_function_association to Product_class

Each Class_function_association refers to exactly one Product_class in the role of product_class. Each Product_class acts as product_class for zero or more Class_function_association objects.

4.3.213 Class_function_association to Product_function

Each Class_function_association refers to exactly one Product_function in the role of product_function. Each Product_function acts as product_function for zero or more Class_function_association objects.

4.3.214 Class_inclusion_association to Multi_language_string

Each Class_inclusion_association refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Class_inclusion_association objects.

NOTE – This assertion is established through string_select.

4.3.215 Class_inclusion_association to Product_class

Each Class_inclusion_association refers to exactly one Product_class in the role of product_class. Each Product_class acts as product_class for zero or more Class_inclusion_association objects.

4.3.216 Class_inclusion_association to Specification_inclusion

Each Class_inclusion_association refers to exactly one Specification_inclusion in the role of inclusion. Each Specification_inclusion acts as inclusion for zero or more Class_inclusion_association objects.

4.3.217 Class_specification_association to Product_class

Each Class_specification_association refers to exactly one Product_class in the role of product_class. Each Product_class acts as product_class for zero or more Class_specification_association objects.

4.3.218 Class_specification_association to Specification

Each Class_specification_association refers to exactly one Specification in the role of specification. Each Specification acts as specification for zero or more Class_specification_association objects.

4.3.219 Classification_attribute to External_document

Each Classification_attribute refers to zero or one External_document in the role of attribute_source. Each External_document acts as attribute_source for zero or more Classification_attribute objects.

NOTE – This assertion is established through external_source_select.

4.3.220 Classification_attribute to External_library_reference

Each Classification_attribute refers to zero or one External_library_reference in the role of attribute_source. Each External_library_reference acts as attribute_source for zero or more Classification_attribute objects.

NOTE – This assertion is established through external_source_select.

4.3.221 Classification_attribute to Multi_language_string

Each Classification_attribute refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Classification_attribute objects.

NOTE – This assertion is established through string_select.

Each Classification_attribute refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Classification_attribute objects.

NOTE – This assertion is established through string_select.

4.3.222 Classification_attribute to Property_value

Each Classification_attribute refers to one or more Property_value objects in the role of attribute_value. Each Property_value acts as attribute_value for zero or more Classification_attribute objects.

4.3.223 Classification_system to External_document

Each Classification_system refers to zero or one External_document in the role of classification_system_source. Each External_document acts as classification_system_source for zero or more Classification_system objects.

NOTE – This assertion is established through external_source_select.

4.3.224 Classification_system to External_library_reference

Each Classification_system refers to zero or one External_library_reference in the role of classification_system_source. Each External_library_reference acts as classification_system_source for zero or more Classification_system objects.

NOTE – This assertion is established through external_source_select.

4.3.225 Classification_system to Multi_language_string

Each Classification_system refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Classification_system objects.

NOTE – This assertion is established through string_select.

Each Classification_system refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Classification_system objects.

NOTE – This assertion is established through string_select.

4.3.226 Closed_profile_usage to Closed_profile

Each Closed_profile_usage refers to exactly one Closed_profile in the role of profile. Each Closed_profile acts as profile for zero or more Closed_profile_usage objects.

4.3.227 Closed_profile_usage to Transformation

Each Closed_profile_usage refers to zero or one Transformation in the role of placement. Each Transformation acts as placement for zero or more Closed_profile_usage objects.

4.3.228 Closed_shell to Face

Each Closed_shell refers to one or more Face objects in the role of shell_boundary. Each Face acts as shell_boundary for zero or more Closed_shell objects.

4.3.229 Coating_layer to Material

Each Coating_layer refers to exactly one Material in the role of used_material. Each Material acts as used_material for zero or more Coating_layer objects.

4.3.230 Coating_layer to Value_with_unit

Each Coating_layer refers to zero or one Value_with_unit in the role of thickness. Each Value_with_unit acts as thickness for zero or more Coating_layer objects.

4.3.231 Coating_layer_relationship to Coating_layer

Each Coating_layer_relationship refers to exactly one Coating_layer in the role of following_layer. Each Coating_layer acts as following_layer for zero or more Coating_layer_relationship objects.

Each Coating_layer_relationship refers to exactly one Coating_layer in the role of preceding_layer. Each Coating_layer acts as preceding_layer for zero or more Coating_layer_relationship objects.

4.3.232 Coating_layer_relationship to Multi_language_string

Each Coating_layer_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Coating_layer_relationship objects.

NOTE – This assertion is established through string_select.

Each Coating_layer_relationship refers to zero or one Multi_language_string in the role of purpose. Each Multi_language_string acts as purpose for zero or more Coating_layer_relationship objects.

NOTE – This assertion is established through string_select.

4.3.233 Collected_item_association to Collection_definition

Each Collected_item_association is relating to exactly one Collection_definition. Each Collection_definition has collected_items defined by two or more Collected_item_association objects.

NOTE – The relating Collection_definition is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.234 Collection_definition to Multi_language_string

Each Collection_definition refers to zero or one Multi_language_string in the role of purpose. Each Multi_language_string acts as purpose for zero or more Collection_definition objects.

NOTE – This assertion is established through string_select.

4.3.235 Compound_datum to Datum_feature

Each Compound_datum is_made_up_by two or more Datum_feature objects. Each Datum_feature is related to zero or more Compound_datum objects.

4.3.236 Compound_feature_in_panel_element to Compound_feature_in_panel

Each Compound_feature_in_panel_element is_element_for exactly one Compound_feature_in_panel. Each Compound_feature_in_panel has element defined by two or more Compound_feature_in_panel_element objects.

4.3.237 Compound_feature_in_panel_element to Process_feature_in_panel

Each Compound_feature_in_panel_element refers to zero or one Process_feature_in_panel in the role of feature. Each Process_feature_in_panel acts as feature for zero or more Compound_feature_in_panel_element objects.

NOTE – This assertion is established through compound_in_panel_select.

4.3.238 Compound_feature_in_panel_element to Transformation

Each Compound_feature_in_panel_element refers to zero or one Transformation in the role of element_placement. Each Transformation acts as element_placement for zero or more Compound_feature_in_panel_element objects.

4.3.239 Compound_feature_in_panel_element to Transition_feature

Each Compound_feature_in_panel_element refers to zero or one Transition_feature in the role of feature. Each Transition_feature acts as feature for zero or more Compound_feature_in_panel_element objects.

NOTE – This assertion is established through compound_in_panel_select.

4.3.240 Compound_feature_in_panel_element_relationship to Compound_feature_in_panel_element

Each Compound_feature_in_panel_element_relationship refers to exactly one Compound_feature_in_panel_element in the role of predecessor. Each Compound_feature_in_panel_element acts as predecessor for zero or more Compound_feature_in_panel_element_relationship objects.

Each Compound_feature_in_panel_element_relationship refers to exactly one Compound_feature_in_panel_element in the role of successor. Each Compound_feature_in_panel_element acts as successor for zero or more Compound_feature_in_panel_element_relationship objects.

4.3.241 Compound_feature_in_solid_element to Compound_feature_in_solid

Each Compound_feature_in_solid_element is_element_for exactly one Compound_feature_in_solid. Each Compound_feature_in_solid has element defined by two or more Compound_feature_in_solid_element objects.

4.3.242 Compound_feature_in_solid_element to Process_feature_in_solid

Each Compound_feature_in_solid_element refers to zero or one Process_feature_in_solid in the role of feature. Each Process_feature_in_solid acts as feature for zero or more Compound_feature_in_solid_element objects.

NOTE – This assertion is established through compound_in_solid_select.

4.3.243 Compound_feature_in_solid_element to Thread_feature

Each Compound_feature_in_solid_element refers to zero or one Thread_feature in the role of feature. Each Thread_feature acts as feature for zero or more Compound_feature_in_solid_element objects.

NOTE – This assertion is established through compound_in_solid_select.

4.3.244 Compound_feature_in_solid_element to Transformation

Each Compound_feature_in_solid_element refers to zero or one Transformation in the role of element_placement. Each Transformation acts as element_placement for zero or more Compound_feature_in_solid_element objects.

4.3.245 Compound_feature_in_solid_element to Transition_feature

Each Compound_feature_in_solid_element refers to zero or one Transition_feature in the role of feature. Each Transition_feature acts as feature for zero or more Compound_feature_in_solid_element objects.

NOTE – This assertion is established through compound_in_solid_select.

4.3.246 Compound_feature_in_solid_element_relationship to Compound_feature_in_solid_element

Each Compound_feature_in_solid_element_relationship refers to exactly one Compound_feature_in_solid_element in the role of predecessor. Each Compound_feature_in_solid_element acts as predecessor for zero or more Compound_feature_in_solid_element_relationship objects.

Each Compound_feature_in_solid_element_relationship refers to exactly one Compound_feature_in_solid_element in the role of successor. Each Compound_feature_in_solid_element acts as successor for zero or more Compound_feature_in_solid_element_relationship objects.

4.3.247 Compound_model to Topologically_connected_set

Each Compound_model refers to zero or more Topologically_connected_set objects in the role of element. Each Topologically_connected_set acts as element for zero or more Compound_model objects.

4.3.248 Concentricity_tolerance to Datum

Each Concentricity_tolerance refers to exactly one Datum in the role of datum. Each Datum acts as datum for zero or more Concentricity_tolerance objects.

4.3.249 Configuration to Class_condition_association

Each Configuration is_solution_for zero or one Class_condition_association. Each Class_condition_association is related to zero or more Configuration objects.

NOTE – This assertion is established through configured_specification_select.

4.3.250 Configuration to Class_specification_association

Each Configuration is_solution_for zero or one Class_specification_association. Each Class_specification_association is related to zero or more Configuration objects.

NOTE – This assertion is established through configured_specification_select.

4.3.251 Configuration to Item_solution

Each Configuration refers to zero or one Item_solution in the role of usage. Each Item_solution acts as usage for zero or more Configuration objects.

NOTE – This assertion is established through configured_item_select.

4.3.252 Configuration to Process_operation

Each Configuration refers to zero or one Process_operation in the role of usage. Each Process_operation acts as usage for zero or more Configuration objects.

NOTE – This assertion is established through configured_item_select.

4.3.253 Conical_bottom to Feature_parameter

Each Conical_bottom refers to exactly one Feature_parameter in the role of tip_angle. Each Feature_parameter acts as tip_angle for zero or more Conical_bottom objects.

Each Conical_bottom refers to zero or one Feature_parameter in the role of tip_radius. Each Feature_parameter acts as tip_radius for zero or more Conical_bottom objects.

4.3.254 Constant_radius_edge_round to Feature_parameter

Each Constant_radius_edge_round refers to zero or one Feature_parameter in the role of first_offset. Each Feature_parameter acts as first_offset for zero or more Constant_radius_edge_round objects.

Each Constant_radius_edge_round refers to exactly one Feature_parameter in the role of radius. Each Feature_parameter acts as radius for zero or more Constant_radius_edge_round objects.

Each Constant_radius_edge_round refers to zero or one Feature_parameter in the role of second_offset. Each Feature_parameter acts as second_offset for zero or more Constant_radius_edge_round objects.

4.3.255 Constant_radius_fillet to Feature_parameter

Each Constant_radius_fillet refers to zero or one Feature_parameter in the role of first_offset. Each Feature_parameter acts as first_offset for zero or more Constant_radius_fillet objects.

Each Constant_radius_fillet refers to exactly one Feature_parameter in the role of radius. Each Feature_parameter acts as radius for zero or more Constant_radius_fillet objects.

Each Constant_radius_fillet refers to zero or one Feature_parameter in the role of second_offset. Each Feature_parameter acts as second_offset for zero or more Constant_radius_fillet objects.

4.3.256 Constructive_geometric_element to Axis_placement

Each Constructive_geometric_element refers to zero or one Axis_placement in the role of element. Each Axis_placement acts as element for zero or more Constructive_geometric_element objects.

NOTE – This assertion is established through constructive_element_select.

4.3.257 Constructive_geometric_element to Curve

Each Constructive_geometric_element refers to zero or one Curve in the role of element. Each Curve acts as element for zero or more Constructive_geometric_element objects.

NOTE – This assertion is established through constructive_element_select.

4.3.258 Constructive_geometric_element to Face

Each Constructive_geometric_element refers to zero or one Face in the role of element. Each Face acts as element for zero or more Constructive_geometric_element objects.

NOTE – This assertion is established through constructive_element_select.

4.3.259 Constructive_geometric_element to Oriented_edge_curve

Each Constructive_geometric_element refers to zero or one Oriented_edge_curve in the role of element. Each Oriented_edge_curve acts as element for zero or more Constructive_geometric_element objects.

NOTE – This assertion is established through constructive_element_select.

4.3.260 Constructive_geometric_element to Point

Each Constructive_geometric_element refers to zero or one Point in the role of element. Each Point acts as element for zero or more Constructive_geometric_element objects.

NOTE – This assertion is established through constructive_element_select.

4.3.261 Constructive_geometric_element to Surface

Each Constructive_geometric_element refers to zero or one Surface in the role of element. Each Surface acts as element for zero or more Constructive_geometric_element objects.

NOTE – This assertion is established through constructive_element_select.

4.3.262 Constructive_geometric_element to Vertex_point

Each Constructive_geometric_element refers to zero or one Vertex_point in the role of element. Each Vertex_point acts as element for zero or more Constructive_geometric_element objects.

NOTE – This assertion is established through constructive_element_select.

4.3.263 Constructive_geometry to Constructive_geometric_element

Each Constructive_geometry refers to one or more Constructive_geometric_element objects in the role of element. Each Constructive_geometric_element has geometry defined by one or more Constructive_geometry objects.

4.3.264 Constructive_geometry_association to Constructive_geometry

Each Constructive_geometry_association refers to zero or one Constructive_geometry in the role of base_geometry. Each Constructive_geometry has associated_model defined by one or more Constructive_geometry_association objects.

NOTE – This assertion is established through constructive_geometry_select.

Each Constructive_geometry_association refers to exactly one Constructive_geometry in the role of constructive_geometry. Each Constructive_geometry has associated_model defined by one or more Constructive_geometry_association objects.

4.3.265 Constructive_geometry_association to Geometric_model

Each Constructive_geometry_association refers to zero or one Geometric_model in the role of base_geometry. Each Geometric_model acts as base_geometry for zero or more Constructive_-geometry_association objects.

NOTE – This assertion is established through constructive_geometry_select.

4.3.266 Constructive_geometry_association to Multi_language_string

Each Constructive_geometry_association refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Constructive_-

geometry_association objects.

NOTE – This assertion is established through string_select.

4.3.267 Contact_ratio to Physical_model

Each Contact_ratio refers to exactly one Physical_model in the role of measuring_reference. Each Physical_model acts as measuring_reference for zero or more Contact_ratio objects.

4.3.268 Context_dependent_appearance_assignment to Drawing_sheet

Each Context_dependent_appearance_assignment refers to zero or one Drawing_sheet in the role of appearance_context. Each Drawing_sheet acts as appearance_context for zero or more Context_dependent_appearance_assignment objects.

NOTE – This assertion is established through appearance_context_select and area_select.

4.3.269 Context_dependent_appearance_assignment to Group

Each Context_dependent_appearance_assignment refers to zero or one Group in the role of appearance_context. Each Group acts as appearance_context for zero or more Context_dependent_appearance_assignment objects.

NOTE – This assertion is established through appearance_context_select and structuring_method_select.

4.3.270 Context_dependent_appearance_assignment to Layer

Each Context_dependent_appearance_assignment refers to zero or one Layer in the role of appearance_context. Each Layer acts as appearance_context for zero or more Context_dependent_appearance_assignment objects.

NOTE – This assertion is established through appearance_context_select and structuring_method_select.

4.3.271 Context_dependent_appearance_assignment to Model_image

Each Context_dependent_appearance_assignment refers to zero or one Model_image in the role of appearance_context. Each Model_image acts as appearance_context for zero or more Context_dependent_appearance_assignment objects.

NOTE – This assertion is established through appearance_context_select.

4.3.272 Context_dependent_appearance_assignment to View

Each Context_dependent_appearance_assignment refers to zero or one View in the role of appearance_context. Each View acts as appearance_context for zero or more Context_dependent_appearance_assignment objects.

NOTE – This assertion is established through appearance_context_select.

4.3.273 Context_dependent_appearance_assignment to View_area

Each Context_dependent_appearance_assignment refers to zero or one View_area in the role of appearance_context. Each View_area acts as appearance_context for zero or more Context_dependent_appearance_assignment objects.

NOTE – This assertion is established through appearance_context_select and area_select.

4.3.274 Contract to Activity

Each Contract refers to zero or more Activity objects in the role of contracted_element. Each Activity acts as contracted_element for zero or more Contract objects.

NOTE – This assertion is established through contracted_element_select.

4.3.275 Contract to Item_version

Each Contract refers to zero or more Item_version objects in the role of contracted_element. Each Item_version acts as contracted_element for zero or more Contract objects.

NOTE – This assertion is established through contracted_element_select.

4.3.276 Csg_model to B_rep_model

Each Csg_model refers to zero or more B_rep_model objects in the role of tree_root_expression. Each B_rep_model acts as tree_root_expression for zero or more Csg_model objects.

NOTE – This assertion is established through csg_select and non_primitive_solid_select.

4.3.277 Csg_model to Boolean_result

Each Csg_model refers to zero or more Boolean_result objects in the role of tree_root_expression. Each Boolean_result acts as tree_root_expression for zero or more Csg_model objects.

NOTE – This assertion is established through csg_select.

4.3.278 Csg_model to Csg_primitive

Each Csg_model refers to zero or more Csg_primitive objects in the role of tree_root_expression. Each Csg_primitive acts as tree_root_expression for zero or more Csg_model objects.

NOTE – This assertion is established through csg_select.

4.3.279 Csg_model to Faceted_b_rep_model

Each Csg_model refers to zero or more Faceted_b_rep_model objects in the role of tree_root_-expression. Each Faceted_b_rep_model acts as tree_root_expression for zero or more Csg_model objects.

NOTE – This assertion is established through csg_select and non_primitive_solid_select.

4.3.280 Csg_model to Solid_replica

Each Csg_model refers to zero or more Solid_replica objects in the role of tree_root_expression. Each Solid_replica acts as tree_root_expression for zero or more Csg_model objects.

NOTE – This assertion is established through csg_select and non_primitive_solid_select.

4.3.281 Csg_model to Swept_face_solid

Each Csg_model refers to zero or more Swept_face_solid objects in the role of tree_root_expression. Each Swept_face_solid acts as tree_root_expression for zero or more Csg_model objects.

NOTE – This assertion is established through csg_select and non_primitive_solid_select.

4.3.282 Curve_set_2d to Curve

Each Curve_set_2d refers to zero or more Curve objects in the role of element. Each Curve acts as element for zero or more Curve_set_2d objects.

NOTE – This assertion is established through point_or_curve_select.

4.3.283 Curve_set_2d to Point

Each Curve_set_2d refers to zero or more Point objects in the role of element. Each Point acts as element for zero or more Curve_set_2d objects.

NOTE – This assertion is established through point_or_curve_select.

4.3.284 Curve_set_3d to Curve

Each Curve_set_3d refers to zero or more Curve objects in the role of element. Each Curve acts as element for zero or more Curve_set_3d objects.

NOTE – This assertion is established through point_or_curve_select.

4.3.285 Curve_set_3d to Point

Each Curve_set_3d refers to zero or more Point objects in the role of element. Each Point acts as element for zero or more Curve_set_3d objects.

NOTE – This assertion is established through point_or_curve_select.

4.3.286 Curved_distance_dimension to Shape_aspect

Each Curved_distance_dimension refers to exactly one Shape_aspect in the role of path. Each Shape_aspect acts as path for zero or more Curved_distance_dimension objects.

4.3.287 Data_environment to Multi_language_string

Each Data_environment refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Data_environment objects.

NOTE – This assertion is established through string_select.

4.3.288 Date_and_person_assignment to Activity

Each Date_and_person_assignment is_applied_to zero or one Activity. Each Activity is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.289 Date_and_person_assignment to Activity_element

Each Date_and_person_assignment is_applied_to zero or one Activity_element. Each Activity_element is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.290 Date_and_person_assignment to Assembly_substitute_relationship

Each Date_and_person_assignment is_applied_to zero or one Assembly_substitute_relationship. Each Assembly_substitute_relationship is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.291 Date_and_person_assignment to Class_category_association

Each Date_and_person_assignment is_applied_to zero or one Class_category_association. Each Class_category_association is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.292 Date_and_person_assignment to Class_condition_association

Each Date_and_person_assignment is_applied_to zero or one Class_condition_association. Each Class_condition_association is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.293 Date_and_person_assignment to Class_function_association

Each Date_and_person_assignment is_applied_to zero or one Class_function_association. Each Class_function_association is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.294 Date_and_person_assignment to Class_inclusion_association

Each Date_and_person_assignment is_applied_to zero or one Class_inclusion_association. Each Class_inclusion_association is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.295 Date_and_person_assignment to Class_specification_association

Each Date_and_person_assignment is_applied_to zero or one Class_specification_association. Each Class_specification_association is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.296 Date_and_person_assignment to Configuration

Each Date_and_person_assignment is_applied_to zero or one Configuration. Each Configuration is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.297 Date_and_person_assignment to Date_and_person_or_organization

Each Date_and_person_assignment refers to exactly one Date_and_person_or_organization in the role of assigned_date_and_person. Each Date_and_person_or_organization acts as assigned_date_-and_person for zero or more Date_and_person_assignment objects.

4.3.298 Date_and_person_assignment to Design_constraint

Each Date_and_person_assignment is_applied_to zero or one Design_constraint. Each Design_-constraint is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_-information_select.

4.3.299 Date_and_person_assignment to Design_discipline_item_definition

Each Date_and_person_assignment is_applied_to zero or one Design_discipline_item_definition. Each Design_discipline_item_definition is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and item_information_-select.

4.3.300 Date_and_person_assignment to Drawing

Each Date_and_person_assignment is_applied_to zero or one Drawing. Each Drawing is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.301 Date_and_person_assignment to External_document

Each Date_and_person_assignment is_applied_to zero or one External_document. Each External_-document is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.302 Date_and_person_assignment to Geometric_model

Each Date_and_person_assignment is_applied_to zero or one Geometric_model. Each Geometric_-model is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.303 Date_and_person_assignment to Item

Each Date_and_person_assignment is_applied_to zero or one Item. Each Item is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.304 Date_and_person_assignment to Item_definition_instance_relationship

Each Date_and_person_assignment is_applied_to zero or one Item_definition_instance_relationship. Each Item_definition_instance_relationship is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_select, and item_relationship_select.

4.3.305 Date_and_person_assignment to Item_definition_relationship

Each Date_and_person_assignment is_applied_to zero or one Item_definition_relationship. Each Item_definition_relationship is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_select, and item_relationship_select.

4.3.306 Date_and_person_assignment to Item_instance

Each Date_and_person_assignment is_applied_to zero or one Item_instance. Each Item_instance is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and item_information_select.

4.3.307 Date_and_person_assignment to Item_instance_relationship

Each Date_and_person_assignment is_applied_to zero or one Item_instance_relationship. Each Item_instance_relationship is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_select, and item_relationship_select.

4.3.308 Date_and_person_assignment to Item_solution

Each Date_and_person_assignment is_applied_to zero or one Item_solution. Each Item_solution is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.309 Date_and_person_assignment to Item_version

Each Date_and_person_assignment is_applied_to zero or one Item_version. Each Item_version is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.310 Date_and_person_assignment to Material

Each Date_and_person_assignment is_applied_to zero or one Material. Each Material is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.311 Date_and_person_assignment to Physical_instance

Each Date_and_person_assignment is_applied_to zero or one Physical_instance. Each Physical_instance is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.312 Date_and_person_assignment to Physical_instance_usage

Each Date_and_person_assignment is_applied_to zero or one Physical_instance_usage. Each Physical_instance_usage is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.313 Date_and_person_assignment to Process_plan

Each Date_and_person_assignment is_applied_to zero or one Process_plan. Each Process_plan is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.314 Date_and_person_assignment to Product_class

Each Date_and_person_assignment is_applied_to zero or one Product_class. Each Product_class is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.315 Date_and_person_assignment to Product_component

Each Date_and_person_assignment is_applied_to zero or one Product_component. Each Product_component is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.316 Date_and_person_assignment to Product_function

Each Date_and_person_assignment is_applied_to zero or one Product_function. Each Product_function is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.317 Date_and_person_assignment to Product_specification

Each Date_and_person_assignment is_applied_to zero or one Product_specification. Each Product_specification is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.318 Date_and_person_assignment to Project

Each Date_and_person_assignment is_applied_to zero or one Project. Each Project is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.319 Date_and_person_assignment to Property

Each Date_and_person_assignment is_applied_to zero or one Property. Each Property is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.320 Date_and_person_assignment to Simulation_information

Each Date_and_person_assignment is_applied_to zero or one Simulation_information. Each Simulation_information is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.321 Date_and_person_assignment to Specification

Each Date_and_person_assignment is_applied_to zero or one Specification. Each Specification is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.322 Date_and_person_assignment to Specification_category

Each Date_and_person_assignment is_applied_to zero or one Specification_category. Each Specification_category is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.323 Date_and_person_assignment to Specification_expression

Each Date_and_person_assignment is_applied_to zero or one Specification_expression. Each Specification_expression is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.324 Date_and_person_assignment to Specification_inclusion

Each Date_and_person_assignment is_applied_to zero or one Specification_inclusion. Each Specification_inclusion is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.325 Date_and_person_assignment to Styled_model

Each Date_and_person_assignment is_applied_to zero or one Styled_model. Each Styled_model is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.326 Date_and_person_assignment to View_area

Each Date_and_person_assignment is_applied_to zero or one View_area. Each View_area is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.327 Date_and_person_assignment to Work_request

Each Date_and_person_assignment is_applied_to zero or one Work_request. Each Work_request is related to zero or more Date_and_person_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.328 Date_and_person_or_organization to Date_time

Each Date_and_person_or_organization refers to exactly one Date_time in the role of actual_date. Each Date_time acts as actual_date for zero or more Date_and_person_or_organization objects.

4.3.329 Date_and_person_or_organization to Organization

Each Date_and_person_or_organization refers to zero or one Organization in the role of person_or_organization. Each Organization acts as person_or_organization for zero or more Date_and_person_or_organization objects.

NOTE – This assertion is established through person_or_organization_select.

4.3.330 Date_and_person_or_organization to Person

Each Date_and_person_or_organization refers to zero or one Person in the role of person_or_organization. Each Person acts as person_or_organization for zero or more Date_and_person_or_organization objects.

NOTE – This assertion is established through person_or_organization_select.

4.3.331 Date_and_person_or_organization to Person_in_organization

Each Date_and_person_or_organization refers to zero or one Person_in_organization in the role of person_or_organization. Each Person_in_organization acts as person_or_organization for zero or more Date_and_person_or_organization objects.

NOTE – This assertion is established through person_or_organization_select.

4.3.332 Date_time_assignment to Activity

Each Date_time_assignment is_applied_to zero or one Activity. Each Activity is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.333 Date_time_assignment to Activity_element

Each Date_time_assignment is_applied_to zero or one Activity_element. Each Activity_element is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.334 Date_time_assignment to Assembly_substitute_relationship

Each Date_time_assignment is_applied_to zero or one Assembly_substitute_relationship. Each Assembly_substitute_relationship is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.335 Date_time_assignment to Class_category_association

Each Date_time_assignment is_applied_to zero or one Class_category_association. Each Class_category_association is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.336 Date_time_assignment to Class_condition_association

Each Date_time_assignment is_applied_to zero or one Class_condition_association. Each Class_condition_association is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.337 Date_time_assignment to Class_function_association

Each Date_time_assignment is_applied_to zero or one Class_function_association. Each Class_function_association is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.338 Date_time_assignment to Class_inclusion_association

Each Date_time_assignment is_applied_to zero or one Class_inclusion_association. Each Class_inclusion_association is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.339 Date_time_assignment to Class_specification_association

Each Date_time_assignment is_applied_to zero or one Class_specification_association. Each Class_specification_association is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.340 Date_time_assignment to Configuration

Each Date_time_assignment is_applied_to zero or one Configuration. Each Configuration is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.341 Date_time_assignment to Date_time

Each Date_time_assignment refers to exactly one Date_time in the role of assigned_date_time. Each Date_time acts as assigned_date_time for zero or more Date_time_assignment objects.

4.3.342 Date_time_assignment to Design_constraint

Each Date_time_assignment is_applied_to zero or one Design_constraint. Each Design_constraint is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.343 Date_time_assignment to Design_discipline_item_definition

Each Date_time_assignment is_applied_to zero or one Design_discipline_item_definition. Each Design_discipline_item_definition is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and item_information_select.

4.3.344 Date_time_assignment to Drawing

Each Date_time_assignment is_applied_to zero or one Drawing. Each Drawing is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.345 Date_time_assignment to External_document

Each Date_time_assignment is_applied_to zero or one External_document. Each External_document is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.346 Date_time_assignment to Geometric_model

Each Date_time_assignment is_applied_to zero or one Geometric_model. Each Geometric_model is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.347 Date_time_assignment to Item

Each Date_time_assignment is_applied_to zero or one Item. Each Item is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.348 Date_time_assignment to Item_definition_instance_relationship

Each Date_time_assignment is_applied_to zero or one Item_definition_instance_relationship. Each Item_definition_instance_relationship is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_select, and item_relationship_select.

4.3.349 Date_time_assignment to Item_definition_relationship

Each Date_time_assignment is_applied_to zero or one Item_definition_relationship. Each Item_definition_relationship is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_select, and item_relationship_select.

4.3.350 Date_time_assignment to Item_instance

Each Date_time_assignment is_applied_to zero or one Item_instance. Each Item_instance is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and item_information_select.

4.3.351 Date_time_assignment to Item_instance_relationship

Each Date_time_assignment is_applied_to zero or one Item_instance_relationship. Each Item_instance_relationship is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_select, and item_relationship_select.

4.3.352 Date_time_assignment to Item_solution

Each Date_time_assignment is_applied_to zero or one Item_solution. Each Item_solution is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.353 Date_time_assignment to Item_version

Each Date_time_assignment is_applied_to zero or one Item_version. Each Item_version is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.354 Date_time_assignment to Material

Each Date_time_assignment is_applied_to zero or one Material. Each Material is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.355 Date_time_assignment to Physical_instance

Each Date_time_assignment is_applied_to zero or one Physical_instance. Each Physical_instance is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.356 Date_time_assignment to Physical_instance_usage

Each Date_time_assignment is_applied_to zero or one Physical_instance_usage. Each Physical_instance_usage is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.357 Date_time_assignment to Process_plan

Each Date_time_assignment is_applied_to zero or one Process_plan. Each Process_plan is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.358 Date_time_assignment to Product_class

Each Date_time_assignment is_applied_to zero or one Product_class. Each Product_class is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.359 Date_time_assignment to Product_component

Each Date_time_assignment is_applied_to zero or one Product_component. Each Product_component is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.360 Date_time_assignment to Product_function

Each Date_time_assignment is_applied_to zero or one Product_function. Each Product_function is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.361 Date_time_assignment to Product_specification

Each Date_time_assignment is_applied_to zero or one Product_specification. Each Product_specification is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.362 Date_time_assignment to Project

Each Date_time_assignment is_applied_to zero or one Project. Each Project is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.363 Date_time_assignment to Property

Each Date_time_assignment is_applied_to zero or one Property. Each Property is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.364 Date_time_assignment to Simulation_information

Each Date_time_assignment is_applied_to zero or one Simulation_information. Each Simulation_information is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.365 Date_time_assignment to Specification

Each Date_time_assignment is_applied_to zero or one Specification. Each Specification is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.366 Date_time_assignment to Specification_category

Each Date_time_assignment is_applied_to zero or one Specification_category. Each Specification_category is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.367 Date_time_assignment to Specification_expression

Each Date_time_assignment is_applied_to zero or one Specification_expression. Each Specification_expression is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.368 Date_time_assignment to Specification_inclusion

Each Date_time_assignment is_applied_to zero or one Specification_inclusion. Each Specification_inclusion is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.369 Date_time_assignment to Styled_model

Each Date_time_assignment is_applied_to zero or one Styled_model. Each Styled_model is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.370 Date_time_assignment to View_area

Each Date_time_assignment is_applied_to zero or one View_area. Each View_area is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.371 Date_time_assignment to Work_request

Each Date_time_assignment is_applied_to zero or one Work_request. Each Work_request is related to zero or more Date_time_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.372 Datum_feature to Datum_target_set

Each Datum_feature is_defined_by zero or one Datum_target_set. Each Datum_target_set is related to zero or more Datum_feature objects.

NOTE – This assertion is established through datum_representation_select.

4.3.373 Datum_feature to Derived_geometry

Each Datum_feature is_defined_by zero or one Derived_geometry. Each Derived_geometry is related to zero or more Datum_feature objects.

NOTE – This assertion is established through datum_representation_select.

4.3.374 Datum_feature to Material_condition

Each Datum_feature is_controlled_by zero or one Material_condition. Each Material_condition is related to zero or more Datum_feature objects.

4.3.375 Datum_feature to Shape_aspect

Each Datum_feature is_defined_by zero or one Shape_aspect. Each Shape_aspect is related to zero or more Datum_feature objects.

NOTE – This assertion is established through datum_representation_select.

4.3.376 Datum_target_set to Datum_target

Each Datum_target_set refers to one or more Datum_target objects in the role of target. Each Datum_target acts as target for zero or more Datum_target_set objects.

4.3.377 Datum_target_set to Multi_language_string

Each Datum_target_set refers to zero or one Multi_language_string in the role of rule_description. Each Multi_language_string acts as rule_description for zero or more Datum_target_set objects.

NOTE – This assertion is established through string_select.

4.3.378 Default_chamfer to Feature_parameter

Each Default_chamfer refers to exactly one Feature_parameter in the role of offset_value. Each Feature_parameter acts as offset_value for zero or more Default_chamfer objects.

4.3.379 Default_chamfer to Multi_language_string

Each Default_chamfer refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Default_chamfer objects.

NOTE – This assertion is established through string_select.

4.3.380 Default_draft_angle to Feature_parameter

Each Default_draft_angle refers to exactly one Feature_parameter in the role of draft_angle_value. Each Feature_parameter acts as draft_angle_value for zero or more Default_draft_angle objects.

4.3.381 Default_draft_angle to Multi_language_string

Each Default_draft_angle refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Default_draft_angle objects.

NOTE – This assertion is established through string_select.

4.3.382 Default_edge_round to Feature_parameter

Each Default_edge_round refers to exactly one Feature_parameter in the role of radius_value. Each Feature_parameter acts as radius_value for zero or more Default_edge_round objects.

4.3.383 Default_edge_round to Multi_language_string

Each Default_edge_round refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Default_edge_round objects.

NOTE – This assertion is established through string_select.

4.3.384 Default_fillet to Feature_parameter

Each Default_fillet refers to exactly one Feature_parameter in the role of radius_value. Each Feature_parameter acts as radius_value for zero or more Default_fillet objects.

4.3.385 Default_fillet to Multi_language_string

Each Default_fillet refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Default_fillet objects.

NOTE – This assertion is established through string_select.

4.3.386 Default_roughness to Multi_language_string

Each Default_roughness refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Default_roughness objects.

NOTE – This assertion is established through string_select.

4.3.387 Default_roughness to Value_with_unit

Each Default_roughness refers to exactly one Value_with_unit in the role of max_value. Each Value_with_unit acts as max_value for zero or more Default_roughness objects.

Each Default_roughness refers to zero or one Value_with_unit in the role of min_value. Each Value_with_unit acts as min_value for zero or more Default_roughness objects.

4.3.388 Default_setting_association to Default_chamfer

Each Default_setting_association refers to zero or more Default_chamfer objects in the role of default_value. Each Default_chamfer acts as default_value for zero or more Default_setting_-association objects.

NOTE – This assertion is established through default_setting_select.

4.3.389 Default_setting_association to Default_draft_angle

Each Default_setting_association refers to zero or more Default_draft_angle objects in the role of default_value. Each Default_draft_angle acts as default_value for zero or more Default_setting_-association objects.

NOTE – This assertion is established through default_setting_select.

4.3.390 Default_setting_association to Default_edge_round

Each Default_setting_association refers to zero or more Default_edge_round objects in the role of default_value. Each Default_edge_round acts as default_value for zero or more Default_setting_association objects.

NOTE – This assertion is established through default_setting_select.

4.3.391 Default_setting_association to Default_fillet

Each Default_setting_association refers to zero or more Default_fillet objects in the role of default_value. Each Default_fillet acts as default_value for zero or more Default_setting_association objects.

NOTE – This assertion is established through default_setting_select.

4.3.392 Default_setting_association to Default_roughness

Each Default_setting_association refers to zero or more Default_roughness objects in the role of default_value. Each Default_roughness acts as default_value for zero or more Default_setting_association objects.

NOTE – This assertion is established through default_setting_select.

4.3.393 Default_setting_association to Default_thickness

Each Default_setting_association refers to zero or more Default_thickness objects in the role of default_value. Each Default_thickness acts as default_value for zero or more Default_setting_association objects.

NOTE – This assertion is established through default_setting_select.

4.3.394 Default_setting_association to Digital_document

Each Default_setting_association refers to zero or one Digital_document in the role of default_context. Each Digital_document acts as default_context for zero or more Default_setting_association objects.

NOTE – This assertion is established through default_context_select.

4.3.395 Default_setting_association to Drawing_sheet

Each Default_setting_association refers to zero or one Drawing_sheet in the role of default_context. Each Drawing_sheet acts as default_context for zero or more Default_setting_association objects.

NOTE – This assertion is established through default_context_select and area_select.

4.3.396 Default_setting_association to General_tolerances

Each Default_setting_association refers to zero or more General_tolerances objects in the role of default_value. Each General_tolerances acts as default_value for zero or more Default_setting_association objects.

NOTE – This assertion is established through default_setting_select.

4.3.397 Default_setting_association to Item_shape

Each Default_setting_association refers to zero or one Item_shape in the role of default_context. Each Item_shape acts as default_context for zero or more Default_setting_association objects.

NOTE – This assertion is established through default_context_select.

4.3.398 Default_setting_association to View_area

Each Default_setting_association refers to zero or one View_area in the role of default_context. Each View_area acts as default_context for zero or more Default_setting_association objects.

NOTE – This assertion is established through default_context_select and area_select.

4.3.399 Default_thickness to Multi_language_string

Each Default_thickness refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Default_thickness objects.

NOTE – This assertion is established through string_select.

4.3.400 Default_thickness to Value_with_unit

Each Default_thickness refers to exactly one Value_with_unit in the role of thickness_value. Each Value_with_unit acts as thickness_value for zero or more Default_thickness objects.

4.3.401 Defined_thread to Feature_parameter

Each Defined_thread refers to exactly one Feature_parameter in the role of crest. Each Feature_parameter acts as crest for zero or more Defined_thread objects.

Each Defined_thread refers to exactly one Feature_parameter in the role of density. Each Feature_parameter acts as density for zero or more Defined_thread objects.

Each Defined_thread refers to exactly one Feature_parameter in the role of major_diameter. Each Feature_parameter acts as major_diameter for zero or more Defined_thread objects.

Each Defined_thread refers to exactly one Feature_parameter in the role of minor_diameter. Each Feature_parameter acts as minor_diameter for zero or more Defined_thread objects.

Each Defined_thread refers to exactly one Feature_parameter in the role of pitch_diameter. Each Feature_parameter acts as pitch_diameter for zero or more Defined_thread objects.

4.3.402 Derived_geometry to Constructive_geometric_element

Each Derived_geometry refers to exactly one Constructive_geometric_element in the role of geometry. Each Constructive_geometric_element acts as geometry for zero or more Derived_geometry objects.

4.3.403 Derived_geometry to Shape_aspect

Each Derived_geometry is_derived_from one or more Shape_aspect objects. Each Shape_aspect is related to zero or more Derived_geometry objects.

4.3.404 Descriptive_specification to Multi_language_string

Each Descriptive_specification refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Descriptive_specification objects.

NOTE – This assertion is established through string_select.

4.3.405 Design_constraint to Multi_language_string

Each Design_constraint refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Design_constraint objects.

NOTE – This assertion is established through string_select.

Each Design_constraint refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Design_constraint objects.

NOTE – This assertion is established through string_select.

4.3.406 Design_constraint to Product_class

Each Design_constraint is_valid_for zero or more Product_class objects. Each Product_class is related to zero or more Design_constraint objects.

4.3.407 Design_constraint_association to Design_constraint

Each Design_constraint_association is_based_on exactly one Design_constraint. Each Design_constraint is related to zero or more Design_constraint_association objects.

4.3.408 Design_constraint_association to Item_solution

Each Design_constraint_association is_constraining zero or one Item_solution. Each Item_solution is related to zero or more Design_constraint_association objects.

NOTE – This assertion is established through solution_base_select.

4.3.409 Design_constraint_association to Multi_language_string

Each Design_constraint_association refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Design_constraint_association objects.

NOTE – This assertion is established through string_select.

4.3.410 Design_constraint_association to Product_component

Each Design_constraint_association is_constraining zero or one Product_component. Each Product_component is related to zero or more Design_constraint_association objects.

NOTE – This assertion is established through solution_base_select.

4.3.411 Design_constraint_relationship to Design_constraint

Each Design_constraint_relationship is related to exactly one Design_constraint. Each Design_constraint is referenced by zero or more Design_constraint_relationship objects as related.

NOTE – The related Design_constraint is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Design_constraint_relationship is relating to exactly one Design_constraint. Each Design_constraint is referenced by zero or more Design_constraint_relationship objects as relating.

NOTE – The relating Design_constraint is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.412 Design_constraint_relationship to Multi_language_string

Each Design_constraint_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Design_constraint_relationship objects.

NOTE – This assertion is established through string_select.

4.3.413 Design_discipline_item_definition to Application_context

Each Design_discipline_item_definition is_relevant_for one or more Application_context objects. Each Application_context is related to zero or more Design_discipline_item_definition objects.

4.3.414 Design_discipline_item_definition to Item_version

Each Design_discipline_item_definition refers to exactly one Item_version in the role of associated_item_version. Each Item_version acts as associated_item_version for zero or more Design_discipline_item_definition objects.

4.3.415 Design_discipline_item_definition to Multi_language_string

Each Design_discipline_item_definition refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Design_discipline_item_definition objects.

NOTE – This assertion is established through string_select.

4.3.416 Detailed_element to Multi_language_string

Each Detailed_element refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Detailed_element objects.

NOTE – This assertion is established through string_select.

4.3.417 Diameter_taper to Feature_parameter

Each Diameter_taper refers to exactly one Feature_parameter in the role of final_diameter. Each Feature_parameter acts as final_diameter for zero or more Diameter_taper objects.

4.3.418 Dimension to Annotation_curve

Each Dimension refers to one or more Annotation_curve objects in the role of composition_curve. Each Annotation_curve acts as composition_curve for zero or more Dimension objects.

4.3.419 Dimension to Dimension_callout

Each Dimension refers to exactly one Dimension_callout in the role of primary_callout. Each Dimension_callout acts as primary_callout for zero or more Dimension objects.

Each Dimension refers to zero or one Dimension_callout in the role of secondary_callout. Each Dimension_callout acts as secondary_callout for zero or more Dimension objects.

4.3.420 Dimension_callout to Geometric_dimension

Each Dimension_callout is_displaying exactly one Geometric_dimension. Each Geometric_dimension is related to zero or more Dimension_callout objects.

4.3.421 Dimension_curve to Annotation_symbol

Each Dimension_curve refers to zero, one, or two Annotation_symbol objects in the role of terminator. Each Annotation_symbol acts as terminator for zero or more Dimension_curve objects.

4.3.422 Dimension_sequence_pair to Dimension

Each Dimension_sequence_pair refers to exactly one Dimension in the role of predecessor. Each Dimension acts as predecessor for zero or more Dimension_sequence_pair objects.

Each Dimension_sequence_pair refers to exactly one Dimension in the role of successor. Each Dimension acts as successor for zero or more Dimension_sequence_pair objects.

4.3.423 Dimension_tolerance to Feature_parameter

Each Dimension_tolerance is_tolerancing zero or one Feature_parameter. Each Feature_parameter is related to zero or more Dimension_tolerance objects.

NOTE – This assertion is established through toleranced_value_select.

4.3.424 Dimension_tolerance to Geometric_dimension

Each Dimension_tolerance is_tolerancing zero or one Geometric_dimension. Each Geometric_dimension is related to zero or more Dimension_tolerance objects.

NOTE – This assertion is established through toleranced_value_select.

4.3.425 Dimension_tolerance to Limits_and_fits

Each Dimension_tolerance is_defined_by zero or one Limits_and_fits. Each Limits_and_fits is related to zero or more Dimension_tolerance objects.

NOTE – This assertion is established through tolerance_definition_select.

4.3.426 Dimension_tolerance to Multi_language_string

Each Dimension_tolerance refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Dimension_tolerance objects.

NOTE – This assertion is established through string_select.

4.3.427 Dimension_tolerance to Plus_minus_tolerance

Each Dimension_tolerance is_defined_by zero or one Plus_minus_tolerance. Each Plus_minus_tolerance is related to zero or more Dimension_tolerance objects.

NOTE – This assertion is established through tolerance_definition_select.

4.3.428 Dimension_tolerance to Value_limit

Each Dimension_tolerance is_defined_by zero or one Value_limit. Each Value_limit is related to zero or more Dimension_tolerance objects.

NOTE – This assertion is established through tolerance_definition_select.

4.3.429 Dimension_tolerance to Value_range

Each Dimension_tolerance is_defined_by zero or one Value_range. Each Value_range is related to zero or more Dimension_tolerance objects.

NOTE – This assertion is established through tolerance_definition_select.

4.3.430 Directed_taper to Feature_parameter

Each Directed_taper refers to exactly one Feature_parameter in the role of draft_angle. Each Feature_parameter acts as draft_angle for zero or more Directed_taper objects.

4.3.431 Document_relationship to External_document

Each Document_relationship is related to exactly one External_document. Each External_document is referenced by zero or more Document_relationship objects as related.

NOTE – The related External_document is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Document_relationship is relating to exactly one External_document. Each External_document is referenced by zero or more Document_relationship objects as relating.

NOTE – The relating External_document is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.432 Document_relationship to Multi_language_string

Each Document_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Document_relationship objects.

NOTE – This assertion is established through string_select.

4.3.433 Document_type to Classification_system

Each Document_type refers to zero or one Classification_system in the role of used_classification_system. Each Classification_system acts as used_classification_system for zero or more Document_type objects.

4.3.434 Drawing to Document_type

Each Drawing refers to zero or more Document_type objects in the role of drawing_type. Each Document_type acts as drawing_type for zero or more Drawing objects.

4.3.435 Drawing to Multi_language_string

Each Drawing refers to zero or one Multi_language_string in the role of title. Each Multi_language_string acts as title for zero or more Drawing objects.

NOTE – This assertion is established through string_select.

4.3.436 Drawing_assignment to Class_function_association

Each Drawing_assignment is_describing zero or one Class_function_association. Each Class_function_association is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.437 Drawing_assignment to Design_constraint

Each Drawing_assignment is_describing zero or one Design_constraint. Each Design_constraint is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.438 Drawing_assignment to Design_discipline_item_definition

Each Drawing_assignment is_describing zero or one Design_discipline_item_definition. Each Design_discipline_item_definition is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select and item_information_select.

4.3.439 Drawing_assignment to Drawing

Each Drawing_assignment refers to exactly one Drawing in the role of drawing. Each Drawing acts as drawing for zero or more Drawing_assignment objects.

4.3.440 Drawing_assignment to Item_definition_instance_relationship

Each Drawing_assignment is_describing zero or one Item_definition_instance_relationship. Each Item_definition_instance_relationship is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select, item_information_select, and item_relationship_select.

4.3.441 Drawing_assignment to Item_definition_relationship

Each Drawing_assignment is_describing zero or one Item_definition_relationship. Each Item_definition_relationship is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select, item_information_select, and item_relationship_select.

4.3.442 Drawing_assignment to Item_instance

Each Drawing_assignment is_describing zero or one Item_instance. Each Item_instance is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select and item_information_select.

4.3.443 Drawing_assignment to Item_instance_relationship

Each Drawing_assignment is_describing zero or one Item_instance_relationship. Each Item_instance_relationship is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select, item_information_select, and item_relationship_select.

4.3.444 Drawing_assignment to Item_solution

Each Drawing_assignment is_describing zero or one Item_solution. Each Item_solution is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.445 Drawing_assignment to Process_operation

Each Drawing_assignment is_describing zero or one Process_operation. Each Process_operation is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select and process_information_select.

4.3.446 Drawing_assignment to Process_plan

Each Drawing_assignment is_describing zero or one Process_plan. Each Process_plan is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select and process_information_select.

4.3.447 Drawing_assignment to Product_class

Each Drawing_assignment is_describing zero or one Product_class. Each Product_class is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.448 Drawing_assignment to Product_component

Each Drawing_assignment is_describing zero or one Product_component. Each Product_component is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.449 Drawing_assignment to Product_function

Each Drawing_assignment is_describing zero or one Product_function. Each Product_function is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.450 Drawing_assignment to Specification

Each Drawing_assignment is_describing zero or one Specification. Each Specification is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.451 Drawing_assignment to Specification_category

Each Drawing_assignment is_describing zero or one Specification_category. Each Specification_category is related to zero or more Drawing_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.452 Drawing_sequence to Drawing

Each Drawing_sequence refers to exactly one Drawing in the role of predecessor. Each Drawing acts as predecessor for zero or more Drawing_sequence objects.

Each Drawing_sequence refers to exactly one Drawing in the role of successor. Each Drawing acts as successor for zero or more Drawing_sequence objects.

4.3.453 Drawing_sheet to Cartesian_coordinate_space_2d

Each Drawing_sheet refers to exactly one Cartesian_coordinate_space_2d in the role of sheet_space. Each Cartesian_coordinate_space_2d acts as sheet_space for zero or more Drawing_sheet objects.

4.3.454 Drawing_sheet to Drawing

Each Drawing_sheet refers to exactly one Drawing in the role of associated_drawing. Each Drawing acts as associated_drawing for zero or more Drawing_sheet objects.

4.3.455 Drawing_sheet to Multi_language_string

Each Drawing_sheet refers to zero or one Multi_language_string in the role of title. Each Multi_language_string acts as title for zero or more Drawing_sheet objects.

NOTE – This assertion is established through string_select.

4.3.456 Edge_loop to Oriented_edge_curve

Each Edge_loop refers to one or more Oriented_edge_curve objects in the role of edge. Each Oriented_edge_curve acts as edge for zero or more Edge_loop objects.

4.3.457 Edge_transition to Oriented_edge_curve

Each Edge_transition refers to exactly one Oriented_edge_curve in the role of edge_1. Each Oriented_edge_curve acts as edge_1 for zero or more Edge_transition objects.

Each Edge_transition refers to exactly one Oriented_edge_curve in the role of edge_2. Each Oriented_edge_curve acts as edge_2 for zero or more Edge_transition objects.

4.3.458 Edge_transition to Vertex_point

Each Edge_transition refers to exactly one Vertex_point in the role of vertex_joint. Each Vertex_point acts as vertex_joint for zero or more Edge_transition objects.

4.3.459 Effectivity to Assembly_substitute_relationship

Each Effectivity is_applied_to zero or more Assembly_substitute_relationship objects. Each Assembly_substitute_relationship is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select and activity_element_and_effectivity_select.

4.3.460 Effectivity to Class_category_association

Each Effectivity is_applied_to zero or more Class_category_association objects. Each Class_category_association is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and configuration_information_select.

4.3.461 Effectivity to Class_condition_association

Each Effectivity is_applied_to zero or more Class_condition_association objects. Each Class_condition_association is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and configuration_information_select.

4.3.462 Effectivity to Class_function_association

Each Effectivity is_applied_to zero or more Class_function_association objects. Each Class_function_association is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and specification_information_select.

4.3.463 Effectivity to Class_inclusion_association

Each Effectivity is_applied_to zero or more Class_inclusion_association objects. Each Class_inclusion_association is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and configuration_information_select.

4.3.464 Effectivity to Class_specification_association

Each Effectivity is_applied_to zero or more Class_specification_association objects. Each Class_specification_association is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and configuration_information_select.

4.3.465 Effectivity to Date_time

Each Effectivity refers to zero or one Date_time in the role of end_definition. Each Date_time acts as end_definition for zero or more Effectivity objects.

NOTE – This assertion is established through period_or_date_select.

Each Effectivity refers to zero or one Date_time in the role of start_definition. Each Date_time acts as start_definition for zero or more Effectivity objects.

NOTE – This assertion is established through event_or_date_select.

4.3.466 Effectivity to Dated_configuration

Each Effectivity is_applied_to zero or more Dated_configuration objects. Each Dated_configuration is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select.

4.3.467 Effectivity to Design_constraint

Each Effectivity is_applied_to zero or more Design_constraint objects. Each Design_constraint is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and specification_information_select.

4.3.468 Effectivity to Drawing

Each Effectivity is_applied_to zero or more Drawing objects. Each Drawing is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select and activity_element_and_effectivity_select.

4.3.469 Effectivity to Event_reference

Each Effectivity refers to zero or one Event_reference in the role of end_definition. Each Event_reference acts as end_definition for zero or more Effectivity objects.

NOTE – This assertion is established through period_or_date_select.

Each Effectivity refers to zero or one Event_reference in the role of start_definition. Each Event_reference acts as start_definition for zero or more Effectivity objects.

NOTE – This assertion is established through event_or_date_select.

4.3.470 Effectivity to External_document

Each Effectivity is_applied_to zero or more External_document objects. Each External_document is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select and activity_element_and_effectivity_select.

4.3.471 Effectivity to Geometric_model

Each Effectivity is_applied_to zero or more Geometric_model objects. Each Geometric_model is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and model_select.

4.3.472 Effectivity to Item

Each Effectivity is_applied_to zero or more Item objects. Each Item is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select and activity_element_and_effectivity_select.

4.3.473 Effectivity to Item_definition_instance_relationship

Each Effectivity is_applied_to zero or more Item_definition_instance_relationship objects. Each Item_definition_instance_relationship is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and item_relationship_select.

4.3.474 Effectivity to Item_definition_relationship

Each Effectivity is_applied_to zero or more Item_definition_relationship objects. Each Item_definition_relationship is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and item_relationship_select.

4.3.475 Effectivity to Item_instance

Each Effectivity is_applied_to zero or more Item_instance objects. Each Item_instance is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select and activity_element_and_effectivity_select.

4.3.476 Effectivity to Item_instance_relationship

Each Effectivity is_applied_to zero or more Item_instance_relationship objects. Each Item_instance_relationship is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and item_relationship_select.

4.3.477 Effectivity to Item_solution

Each Effectivity is_applied_to zero or more Item_solution objects. Each Item_solution is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and specification_information_select.

4.3.478 Effectivity to Item_version

Each Effectivity is_applied_to zero or more Item_version objects. Each Item_version is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select and activity_element_and_effectivity_select.

4.3.479 Effectivity to Material

Each Effectivity is_applied_to zero or more Material objects. Each Material is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select and activity_element_and_effectivity_select.

4.3.480 Effectivity to Multi_language_string

Each Effectivity refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Effectivity objects.

NOTE – This assertion is established through string_select.

4.3.481 Effectivity to Organization

Each Effectivity refers to zero or one Organization in the role of concerned_organization. Each Organization acts as concerned_organization for zero or more Effectivity objects.

4.3.482 Effectivity to Process_plan

Each Effectivity is_applied_to zero or more Process_plan objects. Each Process_plan is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select and activity_element_and_effectivity_-select.

4.3.483 Effectivity to Product_class

Each Effectivity is_applied_to zero or more Product_class objects. Each Product_class is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.484 Effectivity to Product_component

Each Effectivity is_applied_to zero or more Product_component objects. Each Product_component is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.485 Effectivity to Product_function

Each Effectivity is_applied_to zero or more Product_function objects. Each Product_function is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.486 Effectivity to Product_specification

Each Effectivity is_applied_to zero or more Product_specification objects. Each Product_specification is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select and activity_element_and_effectivity_-select.

4.3.487 Effectivity to Property

Each Effectivity is_applied_to zero or more Property objects. Each Property is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select and activity_element_and_effectivity_select.

4.3.488 Effectivity to Security_classification

Each Effectivity is_applied_to zero or more Security_classification objects. Each Security_classification is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select.

4.3.489 Effectivity to Simulation_information

Each Effectivity is_applied_to zero or more Simulation_information objects. Each Simulation_information is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and model_select.

4.3.490 Effectivity to Specification

Each Effectivity is_applied_to zero or more Specification objects. Each Specification is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and specification_information_select.

4.3.491 Effectivity to Specification_category

Each Effectivity is_applied_to zero or more Specification_category objects. Each Specification_category is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and specification_information_select.

4.3.492 Effectivity to Specification_expression

Each Effectivity is_applied_to zero or more Specification_expression objects. Each Specification_expression is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and configuration_information_select.

4.3.493 Effectivity to Specification_inclusion

Each Effectivity is_applied_to zero or more Specification_inclusion objects. Each Specification_inclusion is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_select, and configuration_information_select.

4.3.494 Effectivity to Styled_model

Each Effectivity is_applied_to zero or more Styled_model objects. Each Styled_model is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select, activity_element_and_effectivity_-select, and model_select.

4.3.495 Effectivity to Value_with_unit

Each Effectivity refers to zero or one Value_with_unit in the role of end_definition. Each Value_-with_unit acts as end_definition for zero or more Effectivity objects.

NOTE – This assertion is established through period_or_date_select.

4.3.496 Effectivity to View_area

Each Effectivity is_applied_to zero or more View_area objects. Each View_area is related to zero or more Effectivity objects.

NOTE – This assertion is established through effectivity_select and activity_element_and_effectivity_-select.

4.3.497 Effectivity_relationship to Effectivity

Each Effectivity_relationship is related to exactly one Effectivity. Each Effectivity is referenced by zero or more Effectivity_relationship objects as related.

NOTE – The related Effectivity is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Effectivity_relationship is relating to exactly one Effectivity. Each Effectivity is referenced by zero or more Effectivity_relationship objects as relating.

NOTE – The relating Effectivity is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.498 Effectivity_relationship to Multi_language_string

Each Effectivity_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Effectivity_relationship objects.

NOTE – This assertion is established through string_select.

4.3.499 Element_delivery to Activity_element

Each Element_delivery refers to exactly one Activity_element in the role of deliverable_element. Each Activity_element acts as deliverable_element for zero or more Element_delivery objects.

4.3.500 Element_delivery to Organization

Each Element_delivery refers to exactly one Organization in the role of destination. Each Organization acts as destination for zero or more Element_delivery objects.

4.3.501 Element_delivery to Value_with_unit

Each Element_delivery refers to exactly one Value_with_unit in the role of quantity. Each Value_with_unit acts as quantity for zero or more Element_delivery objects.

4.3.502 Event_reference to Activity

Each Event_reference refers to zero or one Activity in the role of event_context. Each Activity acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.503 Event_reference to Activity_element

Each Event_reference refers to zero or one Activity_element in the role of event_context. Each Activity_element acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.504 Event_reference to Assembly_substitute_relationship

Each Event_reference refers to zero or one Assembly_substitute_relationship in the role of event_context. Each Assembly_substitute_relationship acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.505 Event_reference to Class_category_association

Each Event_reference refers to zero or one Class_category_association in the role of event_context. Each Class_category_association acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.506 Event_reference to Class_condition_association

Each Event_reference refers to zero or one Class_condition_association in the role of event_context. Each Class_condition_association acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.507 Event_reference to Class_function_association

Each Event_reference refers to zero or one Class_function_association in the role of event_context. Each Class_function_association acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.508 Event_reference to Class_inclusion_association

Each Event_reference refers to zero or one Class_inclusion_association in the role of event_context. Each Class_inclusion_association acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.509 Event_reference to Class_specification_association

Each Event_reference refers to zero or one Class_specification_association in the role of event_context. Each Class_specification_association acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.510 Event_reference to Configuration

Each Event_reference refers to zero or one Configuration in the role of event_context. Each Configuration acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.511 Event_reference to Design_constraint

Each Event_reference refers to zero or one Design_constraint in the role of event_context. Each Design_constraint acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.512 Event_reference to Design_discipline_item_definition

Each Event_reference refers to zero or one Design_discipline_item_definition in the role of event_context. Each Design_discipline_item_definition acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and item_information_select.

4.3.513 Event_reference to Drawing

Each Event_reference refers to zero or one Drawing in the role of event_context. Each Drawing acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.514 Event_reference to External_document

Each Event_reference refers to zero or one External_document in the role of event_context. Each External_document acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.515 Event_reference to Geometric_model

Each Event_reference refers to zero or one Geometric_model in the role of event_context. Each Geometric_model acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.516 Event_reference to Item

Each Event_reference refers to zero or one Item in the role of event_context. Each Item acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.517 Event_reference to Item_definition_instance_relationship

Each Event_reference refers to zero or one Item_definition_instance_relationship in the role of event_context. Each Item_definition_instance_relationship acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_select, and item_relationship_select.

4.3.518 Event_reference to Item_definition_relationship

Each Event_reference refers to zero or one Item_definition_relationship in the role of event_context. Each Item_definition_relationship acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_select, and item_relationship_select.

4.3.519 Event_reference to Item_instance

Each Event_reference refers to zero or one Item_instance in the role of event_context. Each Item_instance acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and item_information_-select.

4.3.520 Event_reference to Item_instance_relationship

Each Event_reference refers to zero or one Item_instance_relationship in the role of event_context. Each Item_instance_relationship acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_-select, and item_relationship_select.

4.3.521 Event_reference to Item_solution

Each Event_reference refers to zero or one Item_solution in the role of event_context. Each Item_solution acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and specification_-information_select.

4.3.522 Event_reference to Item_version

Each Event_reference refers to zero or one Item_version in the role of event_context. Each Item_-version acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.523 Event_reference to Material

Each Event_reference refers to zero or one Material in the role of event_context. Each Material acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.524 Event_reference to Physical_instance

Each Event_reference refers to zero or one Physical_instance in the role of event_context. Each Physical_instance acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.525 Event_reference to Physical_instance_usage

Each Event_reference refers to zero or one Physical_instance_usage in the role of event_context. Each Physical_instance_usage acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.526 Event_reference to Process_plan

Each Event_reference refers to zero or one Process_plan in the role of event_context. Each Process_plan acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.527 Event_reference to Product_class

Each Event_reference refers to zero or one Product_class in the role of event_context. Each Product_class acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.528 Event_reference to Product_component

Each Event_reference refers to zero or one Product_component in the role of event_context. Each Product_component acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.529 Event_reference to Product_function

Each Event_reference refers to zero or one Product_function in the role of event_context. Each Product_function acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.530 Event_reference to Product_specification

Each Event_reference refers to zero or one Product_specification in the role of event_context. Each Product_specification acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.531 Event_reference to Project

Each Event_reference refers to zero or one Project in the role of event_context. Each Project acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.532 Event_reference to Property

Each Event_reference refers to zero or one Property in the role of event_context. Each Property acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.533 Event_reference to Simulation_information

Each Event_reference refers to zero or one Simulation_information in the role of event_context. Each Simulation_information acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.534 Event_reference to Specification

Each Event_reference refers to zero or one Specification in the role of event_context. Each Specification acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.535 Event_reference to Specification_category

Each Event_reference refers to zero or one Specification_category in the role of event_context. Each Specification_category acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.536 Event_reference to Specification_expression

Each Event_reference refers to zero or one Specification_expression in the role of event_context. Each Specification_expression acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.537 Event_reference to Specification_inclusion

Each Event_reference refers to zero or one Specification_inclusion in the role of event_context. Each Specification_inclusion acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.538 Event_reference to Styled_model

Each Event_reference refers to zero or one Styled_model in the role of event_context. Each Styled_model acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.539 Event_reference to Value_with_unit

Each Event_reference refers to zero or one Value_with_unit in the role of offset. Each Value_with_unit acts as offset for zero or more Event_reference objects.

4.3.540 Event_reference to View_area

Each Event_reference refers to zero or one View_area in the role of event_context. Each View_area acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.541 Event_reference to Work_request

Each Event_reference refers to zero or one Work_request in the role of event_context. Each Work_request acts as event_context for zero or more Event_reference objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.542 Exclusive_process_operation to Process_operation

Each Exclusive_process_operation refers to two or more Process_operation objects in the role of element. Each Process_operation acts as element for zero or more Exclusive_process_operation objects.

4.3.543 External_document to Document_type

Each External_document refers to zero or one Document_type in the role of document_type. Each Document_type acts as document_type for zero or more External_document objects.

4.3.544 External_document to Multi_language_string

Each External_document refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more External_document objects.

NOTE – This assertion is established through string_select.

Each External_document refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more External_document objects.

NOTE – This assertion is established through string_select.

4.3.545 External_document_assignment to Activity

Each External_document_assignment is_assigned_to zero or one Activity. Each Activity is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.546 External_document_assignment to Approval

Each External_document_assignment is_assigned_to zero or one Approval. Each Approval is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.547 External_document_assignment to Change

Each External_document_assignment is_assigned_to zero or one Change. Each Change is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.548 External_document_assignment to Class_function_association

Each External_document_assignment is_assigned_to zero or one Class_function_association. Each Class_function_association is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, and specification_information_select.

4.3.549 External_document_assignment to Contract

Each External_document_assignment is_assigned_to zero or one Contract. Each Contract is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.550 External_document_assignment to Design_constraint

Each External_document_assignment is_assigned_to zero or one Design_constraint. Each Design_constraint is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, and specification_information_select.

4.3.551 External_document_assignment to Design_discipline_item_definition

Each External_document_assignment is_assigned_to zero or one Design_discipline_item_definition. Each Design_discipline_item_definition is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, and item_information_select.

4.3.552 External_document_assignment to Drawing_sheet

Each External_document_assignment is_assigned_to zero or one Drawing_sheet. Each Drawing_sheet is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select and area_select.

4.3.553 External_document_assignment to External_document

Each External_document_assignment refers to exactly one External_document in the role of assigned_document. Each External_document acts as assigned_document for zero or more External_document_assignment objects.

4.3.554 External_document_assignment to Item_classification

Each External_document_assignment is_assigned_to zero or one Item_classification. Each Item_classification is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.555 External_document_assignment to Item_definition_instance_relationship

Each External_document_assignment is_assigned_to zero or one Item_definition_instance_relationship. Each Item_definition_instance_relationship is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, item_information_select, and item_relationship_select.

4.3.556 External_document_assignment to Item_definition_relationship

Each External_document_assignment is_assigned_to zero or one Item_definition_relationship. Each Item_definition_relationship is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, item_information_select, and item_relationship_select.

4.3.557 External_document_assignment to Item_instance

Each External_document_assignment is_assigned_to zero or one Item_instance. Each Item_instance is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, and item_information_select.

4.3.558 External_document_assignment to Item_instance_relationship

Each External_document_assignment is_assigned_to zero or one Item_instance_relationship. Each Item_instance_relationship is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, item_information_select, and item_relationship_select.

4.3.559 External_document_assignment to Item_solution

Each External_document_assignment is_assigned_to zero or one Item_solution. Each Item_solution is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, and specification_information_select.

4.3.560 External_document_assignment to Material

Each External_document_assignment is_assigned_to zero or one Material. Each Material is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.561 External_document_assignment to Physical_instance

Each External_document_assignment is_assigned_to zero or one Physical_instance. Each Physical_instance is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.562 External_document_assignment to Physical_instance_usage

Each External_document_assignment is_assigned_to zero or one Physical_instance_usage. Each Physical_instance_usage is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.563 External_document_assignment to Process_operation

Each External_document_assignment is_assigned_to zero or one Process_operation. Each Process_operation is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, and process_information_select.

4.3.564 External_document_assignment to Process_operation_resource

Each External_document_assignment is_assigned_to zero or one Process_operation_resource. Each Process_operation_resource is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.565 External_document_assignment to Process_plan

Each External_document_assignment is_assigned_to zero or one Process_plan. Each Process_plan is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, and process_information_select.

4.3.566 External_document_assignment to Product_class

Each External_document_assignment is_assigned_to zero or one Product_class. Each Product_class is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, and specification_information_select.

4.3.567 External_document_assignment to Product_component

Each External_document_assignment is_assigned_to zero or one Product_component. Each Product_component is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, and specification_information_select.

4.3.568 External_document_assignment to Product_function

Each External_document_assignment is_assigned_to zero or one Product_function. Each Product_function is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, and specification_information_select.

4.3.569 External_document_assignment to Product_specification

Each External_document_assignment is_assigned_to zero or one Product_specification. Each Product_specification is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.570 External_document_assignment to Project

Each External_document_assignment is_assigned_to zero or one Project. Each Project is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.571 External_document_assignment to Property

Each External_document_assignment is_assigned_to zero or one Property. Each Property is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.572 External_document_assignment to Retention_period

Each External_document_assignment is_assigned_to zero or one Retention_period. Each Retention_period is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.573 External_document_assignment to Specification

Each External_document_assignment is_assigned_to zero or one Specification. Each Specification is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, and specification_information_select.

4.3.574 External_document_assignment to Specification_category

Each External_document_assignment is_assigned_to zero or one Specification_category. Each Specification_category is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select, general_select, and specification_information_select.

4.3.575 External_document_assignment to Surface_condition

Each External_document_assignment is_assigned_to zero or one Surface_condition. Each Surface_condition is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.576 External_document_assignment to View_area

Each External_document_assignment is_assigned_to zero or one View_area. Each View_area is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select and area_select.

4.3.577 External_document_assignment to Work_order

Each External_document_assignment is_assigned_to zero or one Work_order. Each Work_order is related to zero or more External_document_assignment objects.

NOTE – This assertion is established through external_document_select.

4.3.578 External_model to Cartesian_coordinate_space

Each External_model is_defined_in exactly one Cartesian_coordinate_space. Each Cartesian_coordinate_space is related to zero or more External_model objects.

4.3.579 External_model to Digital_document

Each External_model is_defined_as exactly one Digital_document. Each Digital_document has associated_model_space defined by zero or one External_model.

4.3.580 External_model to Multi_language_string

Each External_model refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more External_model objects.

NOTE – This assertion is established through string_select.

4.3.581 Face to Face_bound

Each Face refers to one or more Face_bound objects in the role of bound. Each Face_bound acts as bound for zero or more Face objects.

4.3.582 Face to Surface

Each Face refers to exactly one Surface in the role of face_geometry. Each Surface acts as face_geometry for zero or more Face objects.

4.3.583 Face_bound to Edge_loop

Each Face_bound refers to zero or one Edge_loop in the role of loop. Each Edge_loop acts as loop for zero or more Face_bound objects.

NOTE – This assertion is established through loop_select.

4.3.584 Face_bound to Vertex_loop

Each Face_bound refers to zero or one Vertex_loop in the role of loop. Each Vertex_loop acts as loop for zero or more Face_bound objects.

NOTE – This assertion is established through loop_select.

4.3.585 Face_transition to Face

Each Face_transition refers to exactly one Face in the role of face_1. Each Face acts as face_1 for zero or more Face_transition objects.

Each Face_transition refers to exactly one Face in the role of face_2. Each Face acts as face_2 for zero or more Face_transition objects.

4.3.586 Face_transition to Oriented_edge_curve

Each Face_transition refers to exactly one Oriented_edge_curve in the role of edge_joint. Each Oriented_edge_curve acts as edge_joint for zero or more Face_transition objects.

4.3.587 Faceted_b_rep to Faceted_closed_shell

Each Faceted_b_rep refers to exactly one Faceted_closed_shell in the role of outer. Each Faceted_closed_shell acts as outer for zero or more Faceted_b_rep objects.

Each Faceted_b_rep refers to zero or more Faceted_closed_shell objects in the role of void. Each Faceted_closed_shell acts as void for zero or more Faceted_b_rep objects.

4.3.588 Faceted_b_rep_model to Faceted_b_rep

Each Faceted_b_rep_model refers to zero or more Faceted_b_rep objects in the role of element. Each Faceted_b_rep acts as element for zero or more Faceted_b_rep_model objects.

4.3.589 Faceted_closed_shell to Faceted_face

Each Faceted_closed_shell refers to four or more Faceted_face objects in the role of shell_boundary. Each Faceted_face acts as shell_boundary for zero or more Faceted_closed_shell objects.

4.3.590 Faceted_face to Plane

Each Faceted_face refers to exactly one Plane in the role of face_geometry. Each Plane acts as face_geometry for zero or more Faceted_face objects.

4.3.591 Faceted_face to Poly_loop

Each Faceted_face refers to one or more Poly_loop objects in the role of bound. Each Poly_loop is_bound_for exactly one Faceted_face.

4.3.592 Feature_definition to Axis_placement

Each Feature_definition refers to zero or one Axis_placement in the role of location_and_orientation. Each Axis_placement acts as location_and_orientation for zero or more Feature_definition objects.

4.3.593 Feature_definition to Cartesian_coordinate_space_3d

Each Feature_definition is_defined_in exactly one Cartesian_coordinate_space_3d. Each Cartesian_coordinate_space_3d is related to zero or more Feature_definition objects.

4.3.594 Feature_occurrence to Multi_language_string

Each Feature_occurrence refers to zero or one Multi_language_string in the role of feature_function. Each Multi_language_string acts as feature_function for zero or more Feature_occurrence objects.

NOTE – This assertion is established through string_select.

4.3.595 Feature_parameter to Value_with_unit

Each Feature_parameter refers to exactly one Value_with_unit in the role of specified_value. Each Value_with_unit acts as specified_value for zero or more Feature_parameter objects.

4.3.596 Feature_placement to Geometric_model

Each Feature_placement refers to zero or one Geometric_model in the role of placement_reference. Each Geometric_model acts as placement_reference for zero or more Feature_placement objects.

NOTE – This assertion is established through placement_reference_select.

4.3.597 Feature_placement to Placed_feature

Each Feature_placement refers to zero or one Placed_feature in the role of placement_reference. Each Placed_feature acts as placement_reference for zero or more Feature_placement objects.

NOTE – This assertion is established through placement_reference_select.

4.3.598 Feature_placement to Transformation

Each Feature_placement refers to exactly one Transformation in the role of transformation. Each Transformation acts as transformation for zero or more Feature_placement objects.

4.3.599 Featured_shape to Item_shape

Each Featured_shape refers to exactly one Item_shape in the role of base_shape. Each Item_shape acts as base_shape for zero or more Featured_shape objects.

4.3.600 Fill_area to Annotation_curve

Each Fill_area refers to one or more Annotation_curve objects in the role of boundary. Each Annotation_curve acts as boundary for zero or more Fill_area objects.

4.3.601 Final_item_solution to Descriptive_specification

Each Final_item_solution refers to zero or more Descriptive_specification objects in the role of final_specification. Each Descriptive_specification acts as final_specification for zero or more Final_item_solution objects.

NOTE – This assertion is established through final_specification_select.

4.3.602 Final_item_solution to Design_discipline_item_definition

Each Final_item_solution refers to zero or more Design_discipline_item_definition objects in the role of final_specification. Each Design_discipline_item_definition acts as final_specification for zero or more Final_item_solution objects.

NOTE – This assertion is established through final_specification_select.

4.3.603 Flat_slot_end_type to Feature_parameter

Each Flat_slot_end_type refers to exactly one Feature_parameter in the role of first_radius. Each Feature_parameter acts as first_radius for zero or more Flat_slot_end_type objects.

Each Flat_slot_end_type refers to exactly one Feature_parameter in the role of second_radius. Each Feature_parameter acts as second_radius for zero or more Flat_slot_end_type objects.

4.3.604 Flat_with_chamfer_bottom to Chamfer_definition

Each Flat_with_chamfer_bottom refers to exactly one Chamfer_definition in the role of chamfer. Each Chamfer_definition acts as chamfer for zero or more Flat_with_chamfer_bottom objects.

4.3.605 Flat_with_radius_bottom to Feature_parameter

Each Flat_with_radius_bottom refers to exactly one Feature_parameter in the role of radius. Each Feature_parameter acts as radius for zero or more Flat_with_radius_bottom objects.

4.3.606 Form_feature_in_panel to Angle_specification

Each Form_feature_in_panel refers to zero or one Angle_specification in the role of base_boundary_defined_by. Each Angle_specification acts as base_boundary_defined_by for zero or more Form_feature_in_panel objects.

NOTE – This assertion is established through profile_or_angle_select.

Each Form_feature_in_panel refers to zero or one Angle_specification in the role of target_boundary_defined_by. Each Angle_specification acts as target_boundary_defined_by for zero or more Form_feature_in_panel objects.

NOTE – This assertion is established through profile_or_angle_select.

4.3.607 Form_feature_in_panel to Closed_profile_usage

Each Form_feature_in_panel refers to zero or one Closed_profile_usage in the role of base_boundary_defined_by. Each Closed_profile_usage acts as base_boundary_defined_by for zero or more Form_feature_in_panel objects.

NOTE – This assertion is established through profile_or_angle_select.

Each Form_feature_in_panel refers to zero or one Closed_profile_usage in the role of target_boundary_defined_by. Each Closed_profile_usage acts as target_boundary_defined_by for zero or more Form_feature_in_panel objects.

NOTE – This assertion is established through profile_or_angle_select.

4.3.608 Form_feature_in_panel to Feature_parameter

Each Form_feature_in_panel refers to exactly one Feature_parameter in the role of base_fillet_radius. Each Feature_parameter acts as base_fillet_radius for zero or more Form_feature_in_panel objects.

Each Form_feature_in_panel refers to exactly one Feature_parameter in the role of target_fillet_radius. Each Feature_parameter acts as target_fillet_radius for zero or more Form_feature_in_panel objects.

4.3.609 Form_feature_in_panel to Shell

Each Form_feature_in_panel refers to zero or one Shell in the role of target_face. Each Shell acts as target_face for zero or more Form_feature_in_panel objects.

NOTE – This assertion is established through general_end_select.

4.3.610 Form_feature_in_panel to Surface

Each Form_feature_in_panel refers to zero or one Surface in the role of target_face. Each Surface acts as target_face for zero or more Form_feature_in_panel objects.

NOTE – This assertion is established through general_end_select.

4.3.611 Function_component_association to Product_component

Each Function_component_association refers to exactly one Product_component in the role of product_component. Each Product_component acts as product_component for zero or more Function_component_association objects.

4.3.612 Function_component_association to Product_function

Each Function_component_association refers to exactly one Product_function in the role of product_function. Each Product_function acts as product_function for zero or more Function_-

component_association objects.

4.3.613 General_boss to General_closed_profile

Each General_boss refers to exactly one General_closed_profile in the role of boundary. Each General_closed_profile acts as boundary for zero or more General_boss objects.

4.3.614 General_bottom_condition to Feature_parameter

Each General_bottom_condition refers to zero or one Feature_parameter in the role of floor_radius. Each Feature_parameter acts as floor_radius for zero or more General_bottom_condition objects.

4.3.615 General_bottom_condition to Shell

Each General_bottom_condition refers to zero or one Shell in the role of floor_face. Each Shell acts as floor_face for zero or more General_bottom_condition objects.

NOTE – This assertion is established through general_end_select.

4.3.616 General_bottom_condition to Surface

Each General_bottom_condition refers to zero or one Surface in the role of floor_face. Each Surface acts as floor_face for zero or more General_bottom_condition objects.

NOTE – This assertion is established through general_end_select.

4.3.617 General_classification to Classification_attribute

Each General_classification refers to zero or more Classification_attribute objects in the role of attribute. Each Classification_attribute has associated_classification defined by one or more General_classification objects.

4.3.618 General_classification to Classification_system

Each General_classification refers to zero or one Classification_system in the role of used_classification_system. Each Classification_system has allowed_class defined by zero or more General_classification objects.

4.3.619 General_classification to External_document

Each General_classification refers to zero or one External_document in the role of classification_source. Each External_document acts as classification_source for zero or more General_classification objects.

NOTE – This assertion is established through external_source_select.

4.3.620 General_classification to External_library_reference

Each General_classification refers to zero or one External_library_reference in the role of classification_source. Each External_library_reference acts as classification_source for zero or more General_classification objects.

NOTE – This assertion is established through external_source_select.

4.3.621 General_classification to Multi_language_string

Each General_classification refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more General_classification objects.

NOTE – This assertion is established through string_select.

4.3.622 General_classification_hierarchy to General_classification

Each General_classification_hierarchy refers to exactly one General_classification in the role of sub_classification. Each General_classification acts as sub_classification for zero or more General_classification_hierarchy objects.

Each General_classification_hierarchy refers to exactly one General_classification in the role of super_classification. Each General_classification acts as super_classification for zero or more General_classification_hierarchy objects.

4.3.623 General_closed_profile to Curve

Each General_closed_profile refers to exactly one Curve in the role of profile_shape. Each Curve acts as profile_shape for zero or more General_closed_profile objects.

4.3.624 General_compound_feature to Feature_definition

Each General_compound_feature refers to zero or more Feature_definition objects in the role of elements. Each Feature_definition acts as elements for zero or more General_compound_feature objects.

NOTE – This assertion is established through feature_definition_select.

4.3.625 General_compound_feature to General_feature

Each General_compound_feature refers to zero or more General_feature objects in the role of elements. Each General_feature acts as elements for zero or more General_compound_feature objects.

NOTE – This assertion is established through feature_definition_select.

4.3.626 General_compound_feature to Thread_feature

Each General_compound_feature refers to zero or more Thread_feature objects in the role of elements. Each Thread_feature acts as elements for zero or more General_compound_feature objects.

NOTE – This assertion is established through feature_definition_select.

4.3.627 General_compound_feature to Transition_feature

Each General_compound_feature refers to zero or more Transition_feature objects in the role of elements. Each Transition_feature acts as elements for zero or more General_compound_feature objects.

NOTE – This assertion is established through feature_definition_select.

4.3.628 General_feature to Axis_placement

Each General_feature refers to zero or one Axis_placement in the role of location_and_orientation. Each Axis_placement acts as location_and_orientation for zero or more General_feature objects.

4.3.629 General_feature to General_feature_parameter

Each General_feature refers to zero or more General_feature_parameter objects in the role of feature_parameter. Each General_feature_parameter acts as feature_parameter for zero or more General_feature objects.

4.3.630 General_feature_parameter to Feature_parameter

Each General_feature_parameter refers to exactly one Feature_parameter in the role of parameter_value. Each Feature_parameter acts as parameter_value for zero or more General_feature_parameter objects.

4.3.631 General_feature_parameter to Multi_language_string

Each General_feature_parameter refers to zero or one Multi_language_string in the role of parameter_role. Each Multi_language_string acts as parameter_role for zero or more General_feature_parameter objects.

NOTE – This assertion is established through string_select.

4.3.632 General_item_definition_instance_relationship to Multi_language_string

Each General_item_definition_instance_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more General_item_definition_instance_relationship objects.

NOTE – This assertion is established through string_select.

Each General_item_definition_instance_relationship refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more General_item_definition_instance_relationship objects.

NOTE – This assertion is established through string_select.

4.3.633 General_item_definition_relationship to Multi_language_string

Each General_item_definition_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more General_item_definition_relationship objects.

NOTE – This assertion is established through string_select.

Each General_item_definition_relationship refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more General_item_definition_relationship objects.

NOTE – This assertion is established through string_select.

4.3.634 General_item_instance_relationship to Multi_language_string

Each General_item_instance_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more General_item_instance_relationship objects.

NOTE – This assertion is established through string_select.

Each General_item_instance_relationship refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more General_item_instance_relationship objects.

NOTE – This assertion is established through string_select.

4.3.635 General_item_property to External_model

Each General_item_property refers to zero or more External_model objects in the role of property_value. Each External_model acts as property_value for zero or more General_item_property objects.

NOTE – This assertion is established through general_property_representation_select.

4.3.636 General_item_property to Geometric_model

Each General_item_property refers to zero or more Geometric_model objects in the role of property_value. Each Geometric_model acts as property_value for zero or more General_item_property objects.

NOTE – This assertion is established through general_property_representation_select.

4.3.637 General_item_property to Multi_language_string

Each General_item_property refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more General_item_property objects.

NOTE – This assertion is established through string_select.

4.3.638 General_item_property to Property_value

Each General_item_property refers to zero or more Property_value objects in the role of property_value. Each Property_value acts as property_value for zero or more General_item_property objects.

NOTE – This assertion is established through general_property_representation_select.

4.3.639 General_item_property to Styled_model

Each General_item_property refers to zero or more Styled_model objects in the role of property_value. Each Styled_model acts as property_value for zero or more General_item_property objects.

NOTE – This assertion is established through general_property_representation_select.

4.3.640 General_open_profile to Curve

Each General_open_profile refers to exactly one Curve in the role of profile_shape. Each Curve acts as profile_shape for zero or more General_open_profile objects.

4.3.641 General_path to Curve

Each General_path refers to exactly one Curve in the role of path_geometry. Each Curve acts as path_geometry for zero or more General_path objects.

4.3.642 General_pattern to Axis_placement

Each General_pattern refers to one or more Axis_placement objects in the role of added_instance. Each Axis_placement acts as added_instance for zero or more General_pattern objects.

4.3.643 General_pocket_in_solid to Profile

Each General_pocket_in_solid refers to exactly one Profile in the role of boundary. Each Profile acts as boundary for zero or more General_pocket_in_solid objects.

4.3.644 General_process_property to External_model

Each General_process_property refers to zero or more External_model objects in the role of property_value. Each External_model acts as property_value for zero or more General_process_property objects.

NOTE – This assertion is established through general_property_representation_select.

4.3.645 General_process_property to Geometric_model

Each General_process_property refers to zero or more Geometric_model objects in the role of property_value. Each Geometric_model acts as property_value for zero or more General_process_property objects.

NOTE – This assertion is established through general_property_representation_select.

4.3.646 General_process_property to Multi_language_string

Each General_process_property refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more General_process_property objects.

NOTE – This assertion is established through string_select.

4.3.647 General_process_property to Property_value

Each General_process_property refers to zero or more Property_value objects in the role of property_value. Each Property_value acts as property_value for zero or more General_process_property objects.

NOTE – This assertion is established through general_property_representation_select.

4.3.648 General_process_property to Styled_model

Each General_process_property refers to zero or more Styled_model objects in the role of property_value. Each Styled_model acts as property_value for zero or more General_process_property objects.

NOTE – This assertion is established through general_property_representation_select.

4.3.649 General_tolerances to External_document

Each General_tolerances refers to zero or one External_document in the role of table_definition. Each External_document acts as table_definition for zero or more General_tolerances objects.

NOTE – This assertion is established through general_tolerance_table_select.

4.3.650 General_tolerances to General_tolerance_table

Each General_tolerances refers to zero or one General_tolerance_table in the role of table_definition. Each General_tolerance_table acts as table_definition for zero or more General_tolerances objects.

NOTE – This assertion is established through general_tolerance_table_select.

4.3.651 General_top_condition to Shell

Each General_top_condition refers to zero or one Shell in the role of top_face. Each Shell acts as top_face for zero or more General_top_condition objects.

NOTE – This assertion is established through general_end_select.

4.3.652 General_top_condition to Surface

Each General_top_condition refers to zero or one Surface in the role of top_face. Each Surface acts as top_face for zero or more General_top_condition objects.

NOTE – This assertion is established through general_end_select.

4.3.653 Geometric_dimension to Multi_language_string

Each Geometric_dimension refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Geometric_dimension objects.

NOTE – This assertion is established through string_select.

4.3.654 Geometric_model to Cartesian_coordinate_space

Each Geometric_model is_defined_in exactly one Cartesian_coordinate_space. Each Cartesian_coordinate_space is related to zero or more Geometric_model objects.

4.3.655 Geometric_model to Multi_language_string

Each Geometric_model refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Geometric_model objects.

NOTE – This assertion is established through string_select.

4.3.656 Geometric_model to Template_instance

Each Geometric_model refers to zero or more Template_instance objects in the role of additional_elements. Each Template_instance acts as additional_elements for zero or more Geometric_model objects.

4.3.657 Geometric_model_relationship to External_model

Each Geometric_model_relationship is related to zero or one External_model. Each External_model is referenced by zero or more Geometric_model_relationship objects as related.

NOTE – The related External_model is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

NOTE – This assertion is established through geometric_or_external_model_select.

Each Geometric_model_relationship is relating to zero or one External_model. Each External_model is referenced by zero or more Geometric_model_relationship objects as relating.

NOTE – The relating External_model is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

NOTE – This assertion is established through geometric_or_external_model_select.

4.3.658 Geometric_model_relationship to Geometric_model

Each Geometric_model_relationship is related to zero or one Geometric_model. Each Geometric_model is referenced by zero or more Geometric_model_relationship objects as related.

NOTE – The related Geometric_model is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

NOTE – This assertion is established through geometric_or_external_model_select.

Each Geometric_model_relationship is relating to zero or one Geometric_model. Each Geometric_model is referenced by zero or more Geometric_model_relationship objects as relating.

NOTE – The relating Geometric_model is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

NOTE – This assertion is established through geometric_or_external_model_select.

4.3.659 Geometric_model_relationship to Multi_language_string

Each Geometric_model_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Geometric_model_relationship objects.

NOTE – This assertion is established through string_select.

4.3.660 Geometric_model_relationship_with_transformation to Transformation

Each Geometric_model_relationship_with_transformation refers to exactly one Transformation in the role of transformation. Each Transformation acts as transformation for zero or more Geometric_model_relationship_with_transformation objects.

4.3.661 Geometric_set_3d to Curve

Each Geometric_set_3d refers to zero or more Curve objects in the role of element. Each Curve acts as element for zero or more Geometric_set_3d objects.

NOTE – This assertion is established through point_or_curve_or_surface_select and point_or_curve_-select.

4.3.662 Geometric_set_3d to Point

Each Geometric_set_3d refers to zero or more Point objects in the role of element. Each Point acts as element for zero or more Geometric_set_3d objects.

NOTE – This assertion is established through point_or_curve_or_surface_select and point_or_curve_-select.

4.3.663 Geometric_set_3d to Surface

Each Geometric_set_3d refers to zero or more Surface objects in the role of element. Each Surface acts as element for zero or more Geometric_set_3d objects.

NOTE – This assertion is established through point_or_curve_or_surface_select.

4.3.664 Geometric_tolerance to Annotation_curve

Each Geometric_tolerance is_applied_to zero or one Annotation_curve. Each Annotation_curve is related to zero or more Geometric_tolerance objects.

NOTE – This assertion is established through placed_element_select and element_with_dimension_-select.

4.3.665 Geometric_tolerance to Constructive_geometric_element

Each Geometric_tolerance is_applied_to zero or one Constructive_geometric_element. Each Constructive_geometric_element is related to zero or more Geometric_tolerance objects.

NOTE – This assertion is established through placed_element_select and element_with_dimension_-select.

4.3.666 Geometric_tolerance to Derived_geometry

Each Geometric_tolerance is_applied_to zero or one Derived_geometry. Each Derived_geometry is related to zero or more Geometric_tolerance objects.

NOTE – This assertion is established through placed_element_select.

4.3.667 Geometric_tolerance to Material_condition

Each Geometric_tolerance is_controlled_by zero or one Material_condition. Each Material_condition is related to zero or more Geometric_tolerance objects.

4.3.668 Geometric_tolerance to Multi_language_string

Each Geometric_tolerance refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Geometric_tolerance objects.

NOTE – This assertion is established through string_select.

Each Geometric_tolerance refers to zero or one Multi_language_string in the role of qualifying_note. Each Multi_language_string acts as qualifying_note for zero or more Geometric_tolerance objects.

NOTE – This assertion is established through string_select.

4.3.669 Geometric_tolerance to Shape_aspect

Each Geometric_tolerance is_applied_to zero or one Shape_aspect. Each Shape_aspect is related to zero or more Geometric_tolerance objects.

NOTE – This assertion is established through placed_element_select and element_with_dimension_select.

4.3.670 Geometric_tolerance_precedence to Geometric_tolerance

Each Geometric_tolerance_precedence refers to exactly one Geometric_tolerance in the role of primary_tolerance. Each Geometric_tolerance acts as primary_tolerance for zero or more Geometric_tolerance_precedence objects.

Each Geometric_tolerance_precedence refers to exactly one Geometric_tolerance in the role of secondary_tolerance. Each Geometric_tolerance acts as secondary_tolerance for zero or more Geometric_tolerance_precedence objects.

4.3.671 Geometrical_relationship to Geometric_model_relationship_with_transformation

Each Geometrical_relationship refers to zero or one Geometric_model_relationship_with_transformation in the role of transformation. Each Geometric_model_relationship_with_transformation acts as transformation for zero or more Geometrical_relationship objects.

NOTE – This assertion is established through transformation_select.

4.3.672 Geometrical_relationship to Multi_language_string

Each Geometrical_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Geometrical_relationship objects.

NOTE – This assertion is established through string_select.

Each Geometrical_relationship refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Geometrical_relationship objects.

NOTE – This assertion is established through string_select.

4.3.673 Geometrical_relationship to Template_instance

Each Geometrical_relationship refers to zero or one Template_instance in the role of transformation. Each Template_instance acts as transformation for zero or more Geometrical_relationship objects.

NOTE – This assertion is established through transformation_select.

4.3.674 Geometrically_bounded_surface_model to Geometric_set_3d

Each Geometrically_bounded_surface_model refers to zero or more Geometric_set_3d objects in the role of element. Each Geometric_set_3d acts as element for zero or more Geometrically_bounded_surface_model objects.

4.3.675 Ground_representation to Geometric_model

Each Ground_representation refers to exactly one Geometric_model in the role of shape_description. Each Geometric_model acts as shape_description for zero or more Ground_representation objects.

4.3.676 Group to Detailed_element

Each Group refers to zero or more Detailed_element objects in the role of element. Each Detailed_element acts as element for zero or more Group objects.

4.3.677 Group to Multi_language_string

Each Group refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Group objects.

NOTE – This assertion is established through string_select.

Each Group refers to zero or one Multi_language_string in the role of purpose. Each Multi_language_string acts as purpose for zero or more Group objects.

NOTE – This assertion is established through string_select.

4.3.678 Group_relationship to Group

Each Group_relationship is related to exactly one Group. Each Group is referenced by zero or more Group_relationship objects as related.

NOTE – The related Group is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Group_relationship is relating to exactly one Group. Each Group is referenced by zero or more Group_relationship objects as relating.

NOTE – The relating Group is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.679 Group_relationship to Multi_language_string

Each Group_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Group_relationship objects.

NOTE – This assertion is established through string_select.

4.3.680 Hardness to Multi_language_string

Each Hardness refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Hardness objects.

NOTE – This assertion is established through string_select.

4.3.681 Hardness to Value_with_unit

Each Hardness refers to zero or one Value_with_unit in the role of depth. Each Value_with_unit acts as depth for zero or more Hardness objects.

Each Hardness refers to exactly one Value_with_unit in the role of hardness_value. Each Value_with_unit acts as hardness_value for zero or more Hardness objects.

4.3.682 Hidden_element_handling to Appearance_assignment

Each Hidden_element_handling refers to zero or one Appearance_assignment in the role of hidden_element_appearance. Each Appearance_assignment acts as hidden_element_appearance for zero or more Hidden_element_handling objects.

4.3.683 Hidden_element_handling to Model_image

Each Hidden_element_handling is_defined_for one or more Model_image objects. Each Model_image has zero or one Hidden_element_handling.

4.3.684 Hole_feature_in_panel to Angle_taper

Each Hole_feature_in_panel refers to zero or one Angle_taper in the role of taper. Each Angle_taper acts as taper for zero or more Hole_feature_in_panel objects.

NOTE – This assertion is established through taper_select.

4.3.685 Hole_feature_in_panel to Closed_profile_usage

Each Hole_feature_in_panel refers to exactly one Closed_profile_usage in the role of hole_profile. Each Closed_profile_usage acts as hole_profile for zero or more Hole_feature_in_panel objects.

4.3.686 Hole_feature_in_panel to Diameter_taper

Each Hole_feature_in_panel refers to zero or one Diameter_taper in the role of taper. Each Diameter_taper acts as taper for zero or more Hole_feature_in_panel objects.

NOTE – This assertion is established through taper_select.

4.3.687 Hole_feature_in_panel to Directed_taper

Each Hole_feature_in_panel refers to zero or one Directed_taper in the role of taper. Each Directed_taper acts as taper for zero or more Hole_feature_in_panel objects.

NOTE – This assertion is established through taper_select.

4.3.688 Hybrid_geometric_model_3d to Detailed_geometric_model_element

Each Hybrid_geometric_model_3d refers to zero or more Detailed_geometric_model_element objects in the role of element. Each Detailed_geometric_model_element acts as element for zero or more Hybrid_geometric_model_3d objects.

4.3.689 Included_feature to Feature_definition

Each Included_feature refers to zero or one Feature_definition in the role of definition. Each Feature_definition acts as definition for zero or more Included_feature objects.

NOTE – This assertion is established through feature_definition_select.

4.3.690 Included_feature to General_feature

Each Included_feature refers to zero or one General_feature in the role of definition. Each General_feature acts as definition for zero or more Included_feature objects.

NOTE – This assertion is established through feature_definition_select.

4.3.691 Included_feature to Thread_feature

Each Included_feature refers to zero or one Thread_feature in the role of definition. Each Thread_feature acts as definition for zero or more Included_feature objects.

NOTE – This assertion is established through feature_definition_select.

4.3.692 Included_feature to Transition_feature

Each Included_feature refers to zero or one Transition_feature in the role of definition. Each Transition_feature acts as definition for zero or more Included_feature objects.

NOTE – This assertion is established through feature_definition_select.

4.3.693 Initial_configuration to Kinematic_mechanism

Each Initial_configuration is_applied_to exactly one Kinematic_mechanism. Each Kinematic_mechanism is related to zero or more Initial_configuration objects.

4.3.694 Initial_configuration to Pair_value

Each Initial_configuration refers to one or more Pair_value objects in the role of initial_pair_values. Each Pair_value acts as initial_pair_values for zero or more Initial_configuration objects.

4.3.695 Instance_placement to Geometric_model_relationship_with_transformation

Each Instance_placement refers to zero or one Geometric_model_relationship_with_transformation in the role of transformation. Each Geometric_model_relationship_with_transformation acts as transformation for zero or more Instance_placement objects.

NOTE – This assertion is established through transformation_select.

4.3.696 Instance_placement to Product_component

Each Instance_placement refers to exactly one Product_component in the role of product_component. Each Product_component acts as product_component for zero or more Instance_placement objects.

4.3.697 Instance_placement to Single_instance

Each Instance_placement refers to exactly one Single_instance in the role of placed_instance. Each Single_instance acts as placed_instance for zero or more Instance_placement objects.

4.3.698 Instance_placement to Template_instance

Each Instance_placement refers to zero or one Template_instance in the role of transformation. Each Template_instance acts as transformation for zero or more Instance_placement objects.

NOTE – This assertion is established through transformation_select.

4.3.699 Interpolated_configuration_sequence to Kinematic_configuration_interpolation

Each Interpolated_configuration_sequence refers to one or more Kinematic_configuration_interpolation objects in the role of interpolation. Each Kinematic_configuration_interpolation acts as interpolation for zero or more Interpolated_configuration_sequence objects.

4.3.700 Item to Multi_language_string

Each Item refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Item objects.

NOTE – This assertion is established through string_select.

Each Item refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Item objects.

NOTE – This assertion is established through string_select.

4.3.701 Item_classification to Item

Each Item_classification refers to zero or more Item objects in the role of associated_item. Each Item acts as associated_item for zero or more Item_classification objects.

4.3.702 Item_classification to Multi_language_string

Each Item_classification refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Item_classification objects.

NOTE – This assertion is established through string_select.

4.3.703 Item_cost_property to Property_value

Each Item_cost_property refers to one or more Property_value objects in the role of cost_property_value. Each Property_value acts as cost_property_value for zero or more Item_cost_property objects.

4.3.704 Item_definition_instance_relationship to Design_discipline_item_definition

Each Item_definition_instance_relationship is relating to exactly one Design_discipline_item_definition. Each Design_discipline_item_definition is referenced by zero or more Item_definition_instance_relationship objects as relating.

NOTE – The relating Design_discipline_item_definition is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.705 Item_definition_instance_relationship to Item_instance

Each Item_definition_instance_relationship is related to exactly one Item_instance. Each Item_instance is referenced by zero or more Item_definition_instance_relationship objects as related.

NOTE – The related Item_instance is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

4.3.706 Item_definition_relationship to Design_discipline_item_definition

Each Item_definition_relationship is related to exactly one Design_discipline_item_definition. Each Design_discipline_item_definition is referenced by zero or more Item_definition_relationship objects as related.

NOTE – The related Design_discipline_item_definition is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Item_definition_relationship is relating to exactly one Design_discipline_item_definition. Each Design_discipline_item_definition is referenced by zero or more Item_definition_relationship objects as relating.

NOTE – The relating Design_discipline_item_definition is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.707 Item_instance to Design_discipline_item_definition

Each Item_instance refers to zero or one Design_discipline_item_definition in the role of definition. Each Design_discipline_item_definition acts as definition for zero or more Item_instance objects.

NOTE – This assertion is established through instance_definition_select.

4.3.708 Item_instance to Multi_language_string

Each Item_instance refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Item_instance objects.

NOTE – This assertion is established through string_select.

4.3.709 Item_instance to Product_specification

Each Item_instance refers to zero or one Product_specification in the role of definition. Each Product_specification acts as definition for zero or more Item_instance objects.

NOTE – This assertion is established through instance_definition_select.

4.3.710 Item_instance_relationship to Item_instance

Each Item_instance_relationship is related to exactly one Item_instance. Each Item_instance is referenced by zero or more Item_instance_relationship objects as related.

NOTE – The related Item_instance is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Item_instance_relationship is relating to exactly one Item_instance. Each Item_instance is referenced by zero or more Item_instance_relationship objects as relating.

NOTE – The relating Item_instance is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.711 Item_property to Design_constraint

Each Item_property refers to zero or one Design_constraint in the role of described_object. Each Design_constraint acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select and specification_property_select.

4.3.712 Item_property to Design_discipline_item_definition

Each Item_property refers to zero or one Design_discipline_item_definition in the role of described_object. Each Design_discipline_item_definition acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select and item_information_select.

4.3.713 Item_property to Feature_definition

Each Item_property refers to zero or one Feature_definition in the role of described_object. Each Feature_definition acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select, shape_information_select, and feature_definition_select.

4.3.714 Item_property to General_feature

Each Item_property refers to zero or one General_feature in the role of described_object. Each General_feature acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select, shape_information_select, and feature_definition_select.

4.3.715 Item_property to Item_definition_instance_relationship

Each Item_property refers to zero or one Item_definition_instance_relationship in the role of described_object. Each Item_definition_instance_relationship acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select, item_information_select, and item_relationship_select.

4.3.716 Item_property to Item_definition_relationship

Each Item_property refers to zero or one Item_definition_relationship in the role of described_-object. Each Item_definition_relationship acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select, item_information_select, and item_relationship_select.

4.3.717 Item_property to Item_instance

Each Item_property refers to zero or one Item_instance in the role of described_object. Each Item_instance acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select and item_information_select.

4.3.718 Item_property to Item_instance_relationship

Each Item_property refers to zero or one Item_instance_relationship in the role of described_-object. Each Item_instance_relationship acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select, item_information_select, and item_relationship_select.

4.3.719 Item_property to Item_shape

Each Item_property refers to zero or one Item_shape in the role of described_object. Each Item_shape acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select, shape_information_select, and shape_select.

4.3.720 Item_property to Item_solution

Each Item_property refers to zero or one Item_solution in the role of described_object. Each Item_solution acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select and specification_property_-select.

4.3.721 Item_property to Physical_instance

Each Item_property refers to zero or one Physical_instance in the role of described_object. Each Physical_instance acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select.

4.3.722 Item_property to Product_class

Each Item_property refers to zero or one Product_class in the role of described_object. Each Product_class acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select and specification_property_select.

4.3.723 Item_property to Product_component

Each Item_property refers to zero or one Product_component in the role of described_object. Each Product_component acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select and specification_property_select.

4.3.724 Item_property to Product_function

Each Item_property refers to zero or one Product_function in the role of described_object. Each Product_function acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select and specification_property_select.

4.3.725 Item_property to Shape_aspect

Each Item_property refers to zero or one Shape_aspect in the role of described_object. Each Shape_aspect acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select, shape_information_select, and shape_select.

4.3.726 Item_property to Shape_aspect_relationship

Each Item_property refers to zero or one Shape_aspect_relationship in the role of described_object. Each Shape_aspect_relationship acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select and shape_information_select.

4.3.727 Item_property to Thread_feature

Each Item_property refers to zero or one Thread_feature in the role of described_object. Each Thread_feature acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select, shape_information_select, and feature_definition_select.

4.3.728 Item_property to Transition_feature

Each Item_property refers to zero or one Transition_feature in the role of described_object. Each Transition_feature acts as described_object for zero or more Item_property objects.

NOTE – This assertion is established through item_property_select, shape_information_select, and feature_definition_select.

4.3.729 Item_quality_property to Multi_language_string

Each Item_quality_property refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Item_quality_property objects.

NOTE – This assertion is established through string_select.

4.3.730 Item_quality_property to Property_value

Each Item_quality_property refers to one or more Property_value objects in the role of quality_property_value. Each Property_value acts as quality_property_value for zero or more Item_quality_property objects.

4.3.731 Item_shape to Design_discipline_item_definition

Each Item_shape refers to zero or one Design_discipline_item_definition in the role of described_object. Each Design_discipline_item_definition acts as described_object for zero or more Item_shape objects.

NOTE – This assertion is established through item_information_select.

4.3.732 Item_shape to Item_definition_instance_relationship

Each Item_shape refers to zero or one Item_definition_instance_relationship in the role of described_object. Each Item_definition_instance_relationship acts as described_object for zero or more Item_shape objects.

NOTE – This assertion is established through item_information_select and item_relationship_select.

4.3.733 Item_shape to Item_definition_relationship

Each Item_shape refers to zero or one Item_definition_relationship in the role of described_object. Each Item_definition_relationship acts as described_object for zero or more Item_shape objects.

NOTE – This assertion is established through item_information_select and item_relationship_select.

4.3.734 Item_shape to Item_instance

Each Item_shape refers to zero or one Item_instance in the role of described_object. Each Item_instance acts as described_object for zero or more Item_shape objects.

NOTE – This assertion is established through item_information_select.

4.3.735 Item_shape to Item_instance_relationship

Each Item_shape refers to zero or one Item_instance_relationship in the role of described_object. Each Item_instance_relationship acts as described_object for zero or more Item_shape objects.

NOTE – This assertion is established through item_information_select and item_relationship_select.

4.3.736 Item_shape to Multi_language_string

Each Item_shape refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Item_shape objects.

NOTE – This assertion is established through string_select.

4.3.737 Item_solution to Item_solution

Each Item_solution refers to zero or one Item_solution in the role of base_element. Each Item_solution acts as base_element for zero or more Item_solution objects.

NOTE – This assertion is established through solution_base_select.

4.3.738 Item_solution to Product_component

Each Item_solution refers to zero or one Product_component in the role of base_element. Each Product_component acts as base_element for zero or more Item_solution objects.

NOTE – This assertion is established through solution_base_select.

4.3.739 Item_solution_relationship to Item_solution

Each Item_solution_relationship is related to exactly one Item_solution. Each Item_solution is referenced by zero or more Item_solution_relationship objects as related.

NOTE – The related Item_solution is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Item_solution_relationship is relating to exactly one Item_solution. Each Item_solution is referenced by zero or more Item_solution_relationship objects as relating.

NOTE – The relating Item_solution is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.740 Item_solution_relationship to Multi_language_string

Each Item_solution_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Item_solution_relationship objects.

NOTE – This assertion is established through string_select.

4.3.741 Item_version to Item

Each Item_version refers to exactly one Item in the role of associated_item. Each Item has associated_version defined by one or more Item_version objects.

4.3.742 Item_version to Multi_language_string

Each Item_version refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Item_version objects.

NOTE – This assertion is established through string_select.

4.3.743 Item_version_relationship to Item_version

Each Item_version_relationship is related to exactly one Item_version. Each Item_version is referenced by zero or more Item_version_relationship objects as related.

NOTE – The related Item_version is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Item_version_relationship is relating to exactly one Item_version. Each Item_version is referenced by zero or more Item_version_relationship objects as relating.

NOTE – The relating Item_version is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.744 Item_version_relationship to Multi_language_string

Each Item_version_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Item_version_relationship objects.

NOTE – This assertion is established through string_select.

4.3.745 Joggle_feature to Feature_parameter

Each Joggle_feature refers to exactly one Feature_parameter in the role of direction_angle_at_path_end. Each Feature_parameter acts as direction_angle_at_path_end for zero or more Joggle_feature objects.

Each Joggle_feature refers to exactly one Feature_parameter in the role of direction_angle_at_path_start. Each Feature_parameter acts as direction_angle_at_path_start for zero or more Joggle_feature objects.

Each Joggle_feature refers to exactly one Feature_parameter in the role of height. Each Feature_parameter acts as height for zero or more Joggle_feature objects.

Each Joggle_feature refers to exactly one Feature_parameter in the role of slope_angle_at_path_end. Each Feature_parameter acts as slope_angle_at_path_end for zero or more Joggle_feature objects.

Each Joggle_feature refers to exactly one Feature_parameter in the role of slope_angle_at_path_start. Each Feature_parameter acts as slope_angle_at_path_start for zero or more Joggle_feature objects.

4.3.746 Joggle_feature to Joggle_termination

Each Joggle_feature refers to exactly one Joggle_termination in the role of termination. Each Joggle_termination acts as termination for zero or more Joggle_feature objects.

4.3.747 Joggle_feature to Linear_path

Each Joggle_feature refers to exactly one Linear_path in the role of defining_path. Each Linear_path acts as defining_path for zero or more Joggle_feature objects.

4.3.748 Kinematic_configuration_definition to Pair_value

Each Kinematic_configuration_definition refers to one or more Pair_value objects in the role of pair_values. Each Pair_value acts as pair_values for zero or more Kinematic_configuration_definition objects.

4.3.749 Kinematic_configuration_definition to Path_approximation

Each Kinematic_configuration_definition refers to exactly one Path_approximation in the role of t_parameter. Each Path_approximation acts as t_parameter for zero or more Kinematic_configuration_definition objects.

4.3.750 Kinematic_configuration_interpolation to Kinematic_configuration_definition

Each Kinematic_configuration_interpolation refers to exactly one Kinematic_configuration_definition in the role of next_definition. Each Kinematic_configuration_definition acts as next_definition for zero or more Kinematic_configuration_interpolation objects.

Each Kinematic_configuration_interpolation refers to exactly one Kinematic_configuration_definition in the role of previous_definition. Each Kinematic_configuration_definition acts as previous_definition for zero or more Kinematic_configuration_interpolation objects.

4.3.751 Kinematic_joint to Kinematic_link

Each Kinematic_joint refers to exactly one Kinematic_link in the role of first_link. Each Kinematic_link acts as first_link for zero or more Kinematic_joint objects.

Each Kinematic_joint refers to exactly one Kinematic_link in the role of second_link. Each Kinematic_link acts as second_link for zero or more Kinematic_joint objects.

4.3.752 Kinematic_joint to Kinematic_pair

Each Kinematic_joint refers to exactly one Kinematic_pair in the role of pair. Each Kinematic_pair has joint defined by exactly one Kinematic_joint.

4.3.753 Kinematic_link to Kinematic_link_representation

Each Kinematic_link refers to exactly one Kinematic_link_representation in the role of representation. Each Kinematic_link_representation acts as representation for zero or more Kinematic_link objects.

4.3.754 Kinematic_link_representation to Geometric_model

Each Kinematic_link_representation refers to exactly one Geometric_model in the role of physical_description. Each Geometric_model acts as physical_description for zero or more Kinematic_link_representation objects.

4.3.755 Kinematic_link_representation to Transformation_to_reference_frame

Each Kinematic_link_representation refers to zero or more Transformation_to_reference_frame objects in the role of additional_reference_frame. Each Transformation_to_reference_frame acts as additional_reference_frame for zero or more Kinematic_link_representation objects.

4.3.756 Kinematic_mechanism to Kinematic_link

Each Kinematic_mechanism refers to exactly one Kinematic_link in the role of base. Each Kinematic_link acts as base for zero or more Kinematic_mechanism objects.

4.3.757 Kinematic_mechanism to Kinematic_structure

Each Kinematic_mechanism refers to exactly one Kinematic_structure in the role of structure_-definition. Each Kinematic_structure is_referenced_by exactly one Kinematic_mechanism.

4.3.758 Kinematic_mechanism to Transformation_to_reference_frame

Each Kinematic_mechanism refers to exactly one Transformation_to_reference_frame in the role of base_placement. Each Transformation_to_reference_frame acts as base_placement for zero or more Kinematic_mechanism objects.

4.3.759 Kinematic_pair to Transformation_to_reference_frame

Each Kinematic_pair refers to exactly one Transformation_to_reference_frame in the role of transformation_item_1. Each Transformation_to_reference_frame acts as transformation_item_1 for zero or more Kinematic_pair objects.

Each Kinematic_pair refers to exactly one Transformation_to_reference_frame in the role of transformation_item_2. Each Transformation_to_reference_frame acts as transformation_item_2 for zero or more Kinematic_pair objects.

4.3.760 Kinematic_structure to Kinematic_joint

Each Kinematic_structure refers to one or more Kinematic_joint objects in the role of joint. Each Kinematic_joint is_referenced_by one or more Kinematic_structure objects.

4.3.761 Kinematic_transformation_result to Interpolated_configuration_sequence

Each Kinematic_transformation_result refers to one or more Interpolated_configuration_sequence objects in the role of result. Each Interpolated_configuration_sequence acts as result for zero or more Kinematic_transformation_result objects.

4.3.762 Kinematic_transformation_result to Kinematic_mechanism

Each Kinematic_transformation_result is_applied_to exactly one Kinematic_mechanism. Each Kinematic_mechanism is related to zero or more Kinematic_transformation_result objects.

4.3.763 Layer to Detailed_element

Each Layer refers to one or more Detailed_element objects in the role of element. Each Detailed_-element acts as element for zero or more Layer objects.

4.3.764 Layer to Multi_language_string

Each Layer refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Layer objects.

NOTE – This assertion is established through string_select.

4.3.765 Leader to Annotation_symbol

Each Leader refers to zero or one Annotation_symbol in the role of terminator. Each Annotation_symbol acts as terminator for zero or more Leader objects.

4.3.766 Light_source to Model_image

Each Light_source refers to one or more Model_image objects in the role of view. Each Model_image acts as view for zero or more Light_source objects.

4.3.767 Linear_path to Axis_placement

Each Linear_path is_defined_by exactly one Axis_placement. Each Axis_placement is related to zero or more Linear_path objects.

4.3.768 Linear_path to Feature_parameter

Each Linear_path refers to exactly one Feature_parameter in the role of distance. Each Feature_parameter acts as distance for zero or more Linear_path objects.

4.3.769 Linear_profile_tolerance to Datum

Each Linear_profile_tolerance refers to zero, one, two, or three Datum objects in the role of datum. Each Datum acts as datum for zero or more Linear_profile_tolerance objects.

4.3.770 Linear_profile_tolerance to Reference_plane

Each Linear_profile_tolerance refers to zero or one Reference_plane in the role of affected_plane. Each Reference_plane acts as affected_plane for zero or more Linear_profile_tolerance objects.

4.3.771 Location_dimension to Annotation_curve

Each Location_dimension refers to zero or one Annotation_curve in the role of origin. Each Annotation_curve acts as origin for zero or more Location_dimension objects.

NOTE – This assertion is established through placed_element_select and element_with_dimension_select.

Each Location_dimension refers to zero or one Annotation_curve in the role of target. Each Annotation_curve acts as target for zero or more Location_dimension objects.

NOTE – This assertion is established through placed_element_select and element_with_dimension_select.

4.3.772 Location_dimension to Constructive_geometric_element

Each Location_dimension refers to zero or one Constructive_geometric_element in the role of origin. Each Constructive_geometric_element acts as origin for zero or more Location_dimension objects.

NOTE – This assertion is established through placed_element_select and element_with_dimension_select.

Each Location_dimension refers to zero or one Constructive_geometric_element in the role of target. Each Constructive_geometric_element acts as target for zero or more Location_dimension objects.

NOTE – This assertion is established through placed_element_select and element_with_dimension_select.

4.3.773 Location_dimension to Derived_geometry

Each Location_dimension refers to zero or one Derived_geometry in the role of origin. Each Derived_geometry acts as origin for zero or more Location_dimension objects.

NOTE – This assertion is established through placed_element_select.

Each Location_dimension refers to zero or one Derived_geometry in the role of target. Each Derived_geometry acts as target for zero or more Location_dimension objects.

NOTE – This assertion is established through placed_element_select.

4.3.774 Location_dimension to Shape_aspect

Each Location_dimension refers to zero or one Shape_aspect in the role of origin. Each Shape_aspect acts as origin for zero or more Location_dimension objects.

NOTE – This assertion is established through placed_element_select and element_with_dimension_select.

Each Location_dimension refers to zero or one Shape_aspect in the role of target. Each Shape_aspect acts as target for zero or more Location_dimension objects.

NOTE – This assertion is established through placed_element_select and element_with_dimension_select.

4.3.775 Locator_feature to Feature_parameter

Each Locator_feature refers to exactly one Feature_parameter in the role of diameter. Each Feature_parameter acts as diameter for zero or more Locator_feature objects.

Each Locator_feature refers to exactly one Feature_parameter in the role of tip_angle. Each Feature_parameter acts as tip_angle for zero or more Locator_feature objects.

Each Locator_feature refers to exactly one Feature_parameter in the role of tip_radius. Each Feature_parameter acts as tip_radius for zero or more Locator_feature objects.

4.3.776 Make_from_relationship to Multi_language_string

Each Make_from_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Make_from_relationship objects.

NOTE – This assertion is established through string_select.

Each Make_from_relationship refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Make_from_relationship objects.

NOTE – This assertion is established through string_select.

4.3.777 Mass to Property_value

Each Mass refers to one or more Property_value objects in the role of mass_property_value. Each Property_value acts as mass_property_value for zero or more Mass objects.

4.3.778 Mated_item_association to Mating_definition

Each Mated_item_association is relating to exactly one Mating_definition. Each Mating_definition has mated_items defined by two or more Mated_item_association objects.

NOTE – The relating Mating_definition is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.779 Mated_item_relationship to Geometric_model_relationship_with_transformation

Each Mated_item_relationship refers to zero or one Geometric_model_relationship_with_transformation in the role of transformation. Each Geometric_model_relationship_with_transformation acts as transformation for zero or more Mated_item_relationship objects.

NOTE – This assertion is established through transformation_select.

4.3.780 Mated_item_relationship to Mated_item_association

Each Mated_item_relationship is related to exactly one Mated_item_association. Each Mated_item_association is referenced by zero or more Mated_item_relationship objects as related.

NOTE – The related Mated_item_association is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Mated_item_relationship is relating to exactly one Mated_item_association. Each Mated_-item_association is referenced by zero or more Mated_item_relationship objects as relating.

NOTE – The relating Mated_item_association is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.781 Mated_item_relationship to Quantified_instance

Each Mated_item_relationship refers to zero or more Quantified_instance objects in the role of mating_material. Each Quantified_instance acts as mating_material for zero or more Mated_-item_relationship objects.

4.3.782 Mated_item_relationship to Shape_aspect_relationship

Each Mated_item_relationship refers to zero or one Shape_aspect_relationship in the role of mated_shape. Each Shape_aspect_relationship acts as mated_shape for zero or more Mated_-item_relationship objects.

4.3.783 Mated_item_relationship to Template_instance

Each Mated_item_relationship refers to zero or one Template_instance in the role of transformation. Each Template_instance acts as transformation for zero or more Mated_item_relationship objects.

NOTE – This assertion is established through transformation_select.

4.3.784 Material to Design_constraint

Each Material refers to zero or one Design_constraint in the role of described_object. Each Design_constraint acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select and specification_property_-select.

4.3.785 Material to Design_discipline_item_definition

Each Material refers to zero or one Design_discipline_item_definition in the role of described_-object. Each Design_discipline_item_definition acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select and item_information_select.

4.3.786 Material to Feature_definition

Each Material refers to zero or one Feature_definition in the role of described_object. Each Feature_definition acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select, shape_information_select, and feature_definition_select.

4.3.787 Material to General_feature

Each Material refers to zero or one General_feature in the role of described_object. Each General_feature acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select, shape_information_select, and feature_definition_select.

4.3.788 Material to Item_definition_instance_relationship

Each Material refers to zero or one Item_definition_instance_relationship in the role of described_object. Each Item_definition_instance_relationship acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select, item_information_select, and item_relationship_select.

4.3.789 Material to Item_definition_relationship

Each Material refers to zero or one Item_definition_relationship in the role of described_object. Each Item_definition_relationship acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select, item_information_select, and item_relationship_select.

4.3.790 Material to Item_instance

Each Material refers to zero or one Item_instance in the role of described_object. Each Item_instance acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select and item_information_select.

4.3.791 Material to Item_instance_relationship

Each Material refers to zero or one Item_instance_relationship in the role of described_object. Each Item_instance_relationship acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select, item_information_select, and item_relationship_select.

4.3.792 Material to Item_shape

Each Material refers to zero or one Item_shape in the role of described_object. Each Item_shape acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select, shape_information_select, and shape_select.

4.3.793 Material to Item_solution

Each Material refers to zero or one Item_solution in the role of described_object. Each Item_solution acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select and specification_property_select.

4.3.794 Material to Material_property

Each Material refers to zero or more Material_property objects in the role of material_property. Each Material_property acts as material_property for zero or more Material objects.

4.3.795 Material to Physical_instance

Each Material refers to zero or one Physical_instance in the role of described_object. Each Physical_instance acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select.

4.3.796 Material to Product_class

Each Material refers to zero or one Product_class in the role of described_object. Each Product_class acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select and specification_property_select.

4.3.797 Material to Product_component

Each Material refers to zero or one Product_component in the role of described_object. Each Product_component acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select and specification_property_select.

4.3.798 Material to Product_function

Each Material refers to zero or one Product_function in the role of described_object. Each Product_function acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select and specification_property_select.

4.3.799 Material to Shape_aspect

Each Material refers to zero or one Shape_aspect in the role of described_object. Each Shape_aspect acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select, shape_information_select, and shape_select.

4.3.800 Material to Shape_aspect_relationship

Each Material refers to zero or one Shape_aspect_relationship in the role of described_object. Each Shape_aspect_relationship acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select and shape_information_select.

4.3.801 Material to Thread_feature

Each Material refers to zero or one Thread_feature in the role of described_object. Each Thread_feature acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select, shape_information_select, and feature_definition_select.

4.3.802 Material to Transition_feature

Each Material refers to zero or one Transition_feature in the role of described_object. Each Transition_feature acts as described_object for zero or more Material objects.

NOTE – This assertion is established through item_property_select, shape_information_select, and feature_definition_select.

4.3.803 Material_property to Material_property_representation

Each Material_property refers to one or more Material_property_representation objects in the role of material_property_value. Each Material_property_representation acts as material_property_value for zero or more Material_property objects.

4.3.804 Material_property to Multi_language_string

Each Material_property refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Material_property objects.

NOTE – This assertion is established through string_select.

4.3.805 Material_property_representation to Data_environment

Each Material_property_representation refers to exactly one Data_environment in the role of environment_condition. Each Data_environment acts as environment_condition for zero or more Material_property_representation objects.

4.3.806 Material_property_representation to Property_value

Each Material_property_representation refers to one or more Property_value objects in the role of property_value. Each Property_value acts as property_value for zero or more Material_property_representation objects.

4.3.807 Material_side_specification to Cartesian_coordinate_space

Each Material_side_specification is_defined_in exactly one Cartesian_coordinate_space. Each Cartesian_coordinate_space is related to zero or more Material_side_specification objects.

4.3.808 Material_side_specification to Point_on_surface

Each Material_side_specification refers to exactly one Point_on_surface in the role of position. Each Point_on_surface acts as position for zero or more Material_side_specification objects.

4.3.809 Model_change to Change

Each Model_change is_describing exactly one Change. Each Change is related to zero or more Model_change objects.

4.3.810 Model_change to Detailed_element

Each Model_change refers to zero or more Detailed_element objects in the role of added_element. Each Detailed_element acts as added_element for zero or more Model_change objects.

Each Model_change refers to zero or more Detailed_element objects in the role of deleted_element. Each Detailed_element acts as deleted_element for zero or more Model_change objects.

4.3.811 Model_change to Geometric_model_relationship

Each Model_change refers to zero or one Geometric_model_relationship in the role of change_description. Each Geometric_model_relationship acts as change_description for zero or more Model_change objects.

NOTE – This assertion is established through model_change_select.

4.3.812 Model_change to Multi_language_string

Each Model_change refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Model_change objects.

NOTE – This assertion is established through string_select.

4.3.813 Model_change to Styled_model_relationship

Each Model_change refers to zero or one Styled_model_relationship in the role of change_description. Each Styled_model_relationship acts as change_description for zero or more Model_change objects.

NOTE – This assertion is established through model_change_select.

4.3.814 Model_change to View_area_relationship

Each Model_change refers to zero or one View_area_relationship in the role of change_description. Each View_area_relationship acts as change_description for zero or more Model_change objects.

NOTE – This assertion is established through model_change_select.

4.3.815 Model_image to External_model

Each Model_image is_displaying zero or one External_model. Each External_model is related to zero or more Model_image objects.

NOTE – This assertion is established through presented_model_select.

4.3.816 Model_image to Multi_language_string

Each Model_image refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Model_image objects.

NOTE – This assertion is established through string_select.

4.3.817 Model_image to Styled_model

Each Model_image is_displaying zero or one Styled_model. Each Styled_model is related to zero or more Model_image objects.

NOTE – This assertion is established through presented_model_select.

4.3.818 Model_image to View

Each Model_image is_presented_in exactly one View. Each View has image defined by zero or one Model_image.

4.3.819 Moments_of_inertia to Cartesian_coordinate_space

Each Moments_of_inertia is_defined_in exactly one Cartesian_coordinate_space. Each Cartesian_coordinate_space is related to zero or more Moments_of_inertia objects.

4.3.820 Moments_of_inertia to Value_with_unit

Each Moments_of_inertia refers to one, two, or three Value_with_unit objects in the role of inertia_value. Each Value_with_unit acts as inertia_value for zero or more Moments_of_inertia objects.

4.3.821 Multi_language_string to String_with_language

Each Multi_language_string refers to one or more String_with_language objects in the role of language_dependent_string. Each String_with_language acts as language_dependent_string for zero or more Multi_language_string objects.

4.3.822 Ngon_profile to Feature_parameter

Each Ngon_profile refers to exactly one Feature_parameter in the role of circumscribed_radius. Each Feature_parameter acts as circumscribed_radius for zero or more Ngon_profile objects.

Each Ngon_profile refers to zero or one Feature_parameter in the role of corner_radius. Each Feature_parameter acts as corner_radius for zero or more Ngon_profile objects.

4.3.823 Organization_in_contract to Contract

Each Organization_in_contract refers to one or more Contract objects in the role of contract. Each Contract has release defined by zero or more Organization_in_contract objects.

4.3.824 Organization_in_contract to Date_and_person_or_organization

Each Organization_in_contract refers to zero or more Date_and_person_or_organization objects in the role of signature. Each Date_and_person_or_organization acts as signature for zero or more Organization_in_contract objects.

4.3.825 Organization_in_contract to Multi_language_string

Each Organization_in_contract refers to zero or one Multi_language_string in the role of role_of_organization. Each Multi_language_string acts as role_of_organization for zero or more Organization_in_contract objects.

NOTE – This assertion is established through string_select.

4.3.826 Organization_in_contract to Organization

Each Organization_in_contract refers to exactly one Organization in the role of contracted_organization. Each Organization acts as contracted_organization for zero or more Organization_in_contract objects.

4.3.827 Organization_relationship to Organization

Each Organization_relationship is related to exactly one Organization. Each Organization is referenced by zero or more Organization_relationship objects as related.

NOTE – The related Organization is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Organization_relationship is relating to exactly one Organization. Each Organization is referenced by zero or more Organization_relationship objects as relating.

NOTE – The relating Organization is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.828 Oriented_edge_curve to B_spline_curve

Each Oriented_edge_curve refers to zero or one B_spline_curve in the role of edge_geometry. Each B_spline_curve acts as edge_geometry for zero or more Oriented_edge_curve objects.

NOTE – This assertion is established through topologically_bounded_curve_select.

4.3.829 Oriented_edge_curve to Circle

Each Oriented_edge_curve refers to zero or one Circle in the role of edge_geometry. Each Circle acts as edge_geometry for zero or more Oriented_edge_curve objects.

NOTE – This assertion is established through topologically_bounded_curve_select.

4.3.830 Oriented_edge_curve to Curve_on_surface

Each Oriented_edge_curve refers to zero or one Curve_on_surface in the role of edge_geometry. Each Curve_on_surface acts as edge_geometry for zero or more Oriented_edge_curve objects.

NOTE – This assertion is established through topologically_bounded_curve_select.

4.3.831 Oriented_edge_curve to Ellipse

Each Oriented_edge_curve refers to zero or one Ellipse in the role of edge_geometry. Each Ellipse acts as edge_geometry for zero or more Oriented_edge_curve objects.

NOTE – This assertion is established through topologically_bounded_curve_select.

4.3.832 Oriented_edge_curve to Hyperbola

Each Oriented_edge_curve refers to zero or one Hyperbola in the role of edge_geometry. Each Hyperbola acts as edge_geometry for zero or more Oriented_edge_curve objects.

NOTE – This assertion is established through topologically_bounded_curve_select.

4.3.833 Oriented_edge_curve to Line

Each Oriented_edge_curve refers to zero or one Line in the role of edge_geometry. Each Line acts as edge_geometry for zero or more Oriented_edge_curve objects.

NOTE – This assertion is established through topologically_bounded_curve_select.

4.3.834 Oriented_edge_curve to Parabola

Each Oriented_edge_curve refers to zero or one Parabola in the role of edge_geometry. Each Parabola acts as edge_geometry for zero or more Oriented_edge_curve objects.

NOTE – This assertion is established through topologically_bounded_curve_select.

4.3.835 Oriented_edge_curve to Polyline

Each Oriented_edge_curve refers to zero or one Polyline in the role of edge_geometry. Each Polyline acts as edge_geometry for zero or more Oriented_edge_curve objects.

NOTE – This assertion is established through topologically_bounded_curve_select.

4.3.836 Oriented_edge_curve to Vertex_point

Each Oriented_edge_curve refers to exactly one Vertex_point in the role of edge_end. Each Vertex_point acts as edge_end for zero or more Oriented_edge_curve objects.

Each Oriented_edge_curve refers to exactly one Vertex_point in the role of edge_start. Each Vertex_point acts as edge_start for zero or more Oriented_edge_curve objects.

4.3.837 Pair_value to Kinematic_pair

Each Pair_value is_applied_to exactly one Kinematic_pair. Each Kinematic_pair is related to zero or more Pair_value objects.

4.3.838 Panel_hole_with_barring to Closed_profile_usage

Each Panel_hole_with_barring refers to exactly one Closed_profile_usage in the role of barring_profile. Each Closed_profile_usage acts as barring_profile for zero or more Panel_hole_with_barring objects.

4.3.839 Parallelism_tolerance to Datum

Each Parallelism_tolerance refers to exactly one Datum in the role of datum. Each Datum acts as datum for zero or more Parallelism_tolerance objects.

4.3.840 Parallelism_tolerance to Reference_plane

Each Parallelism_tolerance refers to zero or one Reference_plane in the role of affected_plane. Each Reference_plane acts as affected_plane for zero or more Parallelism_tolerance objects.

4.3.841 Partial_circular_path to Feature_parameter

Each Partial_circular_path refers to exactly one Feature_parameter in the role of sweep_angle. Each Feature_parameter acts as sweep_angle for zero or more Partial_circular_path objects.

4.3.842 Partial_circular_profile to Feature_parameter

Each Partial_circular_profile refers to exactly one Feature_parameter in the role of radius. Each Feature_parameter acts as radius for zero or more Partial_circular_profile objects.

Each Partial_circular_profile refers to exactly one Feature_parameter in the role of sweep_angle. Each Feature_parameter acts as sweep_angle for zero or more Partial_circular_profile objects.

4.3.843 Path to Cartesian_coordinate_space

Each Path is_defined_in exactly one Cartesian_coordinate_space. Each Cartesian_coordinate_space is related to zero or more Path objects.

4.3.844 Perpendicularity_tolerance to Datum

Each Perpendicularity_tolerance refers to exactly one Datum in the role of datum. Each Datum acts as datum for zero or more Perpendicularity_tolerance objects.

4.3.845 Perpendicularity_tolerance to Reference_plane

Each Perpendicularity_tolerance refers to zero or one Reference_plane in the role of affected_plane. Each Reference_plane acts as affected_plane for zero or more Perpendicularity_tolerance objects.

4.3.846 Person_in_organization to Organization

Each Person_in_organization refers to exactly one Organization in the role of organization. Each Organization acts as organization for zero or more Person_in_organization objects.

4.3.847 Person_in_organization to Person

Each Person_in_organization refers to exactly one Person in the role of person. Each Person has organization defined by one or more Person_in_organization objects.

4.3.848 Person_or_organization_assignment to Activity

Each Person_or_organization_assignment is_applied_to zero or one Activity. Each Activity is related to exactly one Person_or_organization_assignment having the role 'custodian' and to zero or more Person_or_organization objects having other roles.

NOTE – This assertion is established through general_organizational_data_select.

4.3.849 Person_or_organization_assignment to Activity_element

Each Person_or_organization_assignment is_applied_to zero or one Activity_element. Each Activity_element is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.850 Person_or_organization_assignment to Assembly_substitute_relationship

Each Person_or_organization_assignment is_applied_to zero or one Assembly_substitute_relationship. Each Assembly_substitute_relationship is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.851 Person_or_organization_assignment to Class_category_association

Each Person_or_organization_assignment is_applied_to zero or one Class_category_association. Each Class_category_association is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.852 Person_or_organization_assignment to Class_condition_association

Each Person_or_organization_assignment is_applied_to zero or one Class_condition_association. Each Class_condition_association is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.853 Person_or_organization_assignment to Class_function_association

Each Person_or_organization_assignment is_applied_to zero or one Class_function_association. Each Class_function_association is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.854 Person_or_organization_assignment to Class_inclusion_association

Each Person_or_organization_assignment is_applied_to zero or one Class_inclusion_association. Each Class_inclusion_association is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.855 Person_or_organization_assignment to Class_specification_association

Each Person_or_organization_assignment is_applied_to zero or one Class_specification_association. Each Class_specification_association is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.856 Person_or_organization_assignment to Configuration

Each Person_or_organization_assignment is_applied_to zero or one Configuration. Each Configuration is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.857 Person_or_organization_assignment to Design_constraint

Each Person_or_organization_assignment is_applied_to zero or one Design_constraint. Each Design_constraint is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.858 Person_or_organization_assignment to Design_discipline_item_definition

Each Person_or_organization_assignment is_applied_to zero or one Design_discipline_item_definition. Each Design_discipline_item_definition is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and item_information_-select.

4.3.859 Person_or_organization_assignment to Drawing

Each Person_or_organization_assignment is_applied_to zero or one Drawing. Each Drawing is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.860 Person_or_organization_assignment to External_document

Each Person_or_organization_assignment is_applied_to zero or one External_document. Each External_document is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.861 Person_or_organization_assignment to Geometric_model

Each Person_or_organization_assignment is_applied_to zero or one Geometric_model. Each Geometric_model is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.862 Person_or_organization_assignment to Item

Each Person_or_organization_assignment is_applied_to zero or one Item. Each Item is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.863 Person_or_organization_assignment to Item_definition_instance_relationship

Each Person_or_organization_assignment is_applied_to zero or one Item_definition_instance_relationship. Each Item_definition_instance_relationship is related to zero or more Person_or_organization_-assignment objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_-select, and item_relationship_select.

4.3.864 Person_or_organization_assignment to Item_definition_relationship

Each Person_or_organization_assignment is_applied_to zero or one Item_definition_relationship. Each Item_definition_relationship is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_select, and item_relationship_select.

4.3.865 Person_or_organization_assignment to Item_instance

Each Person_or_organization_assignment is_applied_to zero or one Item_instance. Each Item_instance is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and item_information_select.

4.3.866 Person_or_organization_assignment to Item_instance_relationship

Each Person_or_organization_assignment is_applied_to zero or one Item_instance_relationship. Each Item_instance_relationship is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_select, and item_relationship_select.

4.3.867 Person_or_organization_assignment to Item_solution

Each Person_or_organization_assignment is_applied_to zero or one Item_solution. Each Item_solution is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.868 Person_or_organization_assignment to Item_version

Each Person_or_organization_assignment is_applied_to zero or one Item_version. Each Item_version is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.869 Person_or_organization_assignment to Material

Each Person_or_organization_assignment is_applied_to zero or one Material. Each Material is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.870 Person_or_organization_assignment to Organization

Each Person_or_organization_assignment refers to zero or one Organization in the role of assigned_person_or_organization. Each Organization acts as assigned_person_or_organization for zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through person_or_organization_select.

4.3.871 Person_or_organization_assignment to Person

Each Person_or_organization_assignment refers to zero or one Person in the role of assigned_person_or_organization. Each Person acts as assigned_person_or_organization for zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through person_or_organization_select.

4.3.872 Person_or_organization_assignment to Person_in_organization

Each Person_or_organization_assignment refers to zero or one Person_in_organization in the role of assigned_person_or_organization. Each Person_in_organization acts as assigned_person_or_organization for zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through person_or_organization_select.

4.3.873 Person_or_organization_assignment to Physical_instance

Each Person_or_organization_assignment is_applied_to zero or one Physical_instance. Each Physical_instance is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.874 Person_or_organization_assignment to Physical_instance_usage

Each Person_or_organization_assignment is_applied_to zero or one Physical_instance_usage. Each Physical_instance_usage is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.875 Person_or_organization_assignment to Process_plan

Each Person_or_organization_assignment is_applied_to zero or one Process_plan. Each Process_plan is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.876 Person_or_organization_assignment to Product_class

Each Person_or_organization_assignment is_applied_to zero or one Product_class. Each Product_class is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.877 Person_or_organization_assignment to Product_component

Each Person_or_organization_assignment is_applied_to zero or one Product_component. Each Product_component is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.878 Person_or_organization_assignment to Product_function

Each Person_or_organization_assignment is_applied_to zero or one Product_function. Each Product_function is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.879 Person_or_organization_assignment to Product_specification

Each Person_or_organization_assignment is_applied_to zero or one Product_specification. Each Product_specification is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.880 Person_or_organization_assignment to Project

Each Person_or_organization_assignment is_applied_to zero or one Project. Each Project is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.881 Person_or_organization_assignment to Property

Each Person_or_organization_assignment is_applied_to zero or one Property. Each Property is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.882 Person_or_organization_assignment to Simulation_information

Each Person_or_organization_assignment is_applied_to zero or one Simulation_information. Each Simulation_information is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.883 Person_or_organization_assignment to Specification

Each Person_or_organization_assignment is_applied_to zero or one Specification. Each Specification is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.884 Person_or_organization_assignment to Specification_category

Each Person_or_organization_assignment is_applied_to zero or one Specification_category. Each Specification_category is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.885 Person_or_organization_assignment to Specification_expression

Each Person_or_organization_assignment is_applied_to zero or one Specification_expression. Each Specification_expression is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.886 Person_or_organization_assignment to Specification_inclusion

Each Person_or_organization_assignment is_applied_to zero or one Specification_inclusion. Each Specification_inclusion is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.887 Person_or_organization_assignment to Styled_model

Each Person_or_organization_assignment is_applied_to zero or one Styled_model. Each Styled_model is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.888 Person_or_organization_assignment to View_area

Each Person_or_organization_assignment is_applied_to zero or one View_area. Each View_area is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.889 Person_or_organization_assignment to Work_request

Each Person_or_organization_assignment is_applied_to zero or one Work_request. Each Work_request is related to zero or more Person_or_organization_assignment objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.890 Physical_instance to Item_version

Each Physical_instance is_based_on zero or one Item_version. Each Item_version is related to zero or more Physical_instance objects.

NOTE – This assertion is established through physical_instance_definition_select.

4.3.891 Physical_instance to Multi_language_string

Each Physical_instance refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Physical_instance objects.

NOTE – This assertion is established through string_select.

4.3.892 Physical_instance to Product_specification

Each Physical_instance is_based_on zero or one Product_specification. Each Product_specification is related to zero or more Physical_instance objects.

NOTE – This assertion is established through physical_instance_definition_select.

4.3.893 Physical_instance_usage to Item_instance

Each Physical_instance_usage refers to exactly one Item_instance in the role of role. Each Item_instance acts as role for zero or more Physical_instance_usage objects.

4.3.894 Physical_instance_usage to Physical_instance

Each Physical_instance_usage refers to exactly one Physical_instance in the role of physical_assembly. Each Physical_instance acts as physical_assembly for zero or more Physical_instance_usage objects.

Each Physical_instance_usage refers to exactly one Physical_instance in the role of physical_component. Each Physical_instance acts as physical_component for zero or more Physical_instance_usage objects.

4.3.895 Placed_feature to Feature_definition

Each Placed_feature refers to exactly one Feature_definition in the role of definition. Each Feature_definition acts as definition for zero or more Placed_feature objects.

4.3.896 Placed_feature to Feature_placement

Each Placed_feature is_placed_by exactly one Feature_placement. Each Feature_placement is_placing exactly one Placed_feature.

4.3.897 Placed_target to Axis_placement

Each Placed_target refers to exactly one Axis_placement in the role of parameter_reference. Each Axis_placement acts as parameter_reference for zero or more Placed_target objects.

4.3.898 Placed_target to Cartesian_coordinate_space

Each Placed_target is_defined_in exactly one Cartesian_coordinate_space. Each Cartesian_coordinate_space is related to zero or more Placed_target objects.

4.3.899 Planar_bottom_condition to Feature_parameter

Each Planar_bottom_condition refers to zero or one Feature_parameter in the role of floor_radius. Each Feature_parameter acts as floor_radius for zero or more Planar_bottom_condition objects.

4.3.900 Pocket_feature_in_solid to Angle_taper

Each Pocket_feature_in_solid refers to zero or one Angle_taper in the role of taper. Each Angle_taper acts as taper for zero or more Pocket_feature_in_solid objects.

NOTE – This assertion is established through taper_select.

4.3.901 Pocket_feature_in_solid to Diameter_taper

Each Pocket_feature_in_solid refers to zero or one Diameter_taper in the role of taper. Each Diameter_taper acts as taper for zero or more Pocket_feature_in_solid objects.

NOTE – This assertion is established through taper_select.

4.3.902 Pocket_feature_in_solid to Directed_taper

Each Pocket_feature_in_solid refers to zero or one Directed_taper in the role of taper. Each Directed_taper acts as taper for zero or more Pocket_feature_in_solid objects.

NOTE – This assertion is established through taper_select.

4.3.903 Pocket_feature_in_solid to Linear_path

Each Pocket_feature_in_solid refers to exactly one Linear_path in the role of sweep_path. Each Linear_path acts as sweep_path for zero or more Pocket_feature_in_solid objects.

4.3.904 Pocket_feature_in_solid to Pocket_bottom_condition

Each Pocket_feature_in_solid refers to exactly one Pocket_bottom_condition in the role of bottom_condition. Each Pocket_bottom_condition acts as bottom_condition for zero or more Pocket_feature_in_solid objects.

4.3.905 Pocket_feature_in_solid to Transformation

Each Pocket_feature_in_solid refers to zero or one Transformation in the role of profile_placement. Each Transformation acts as profile_placement for zero or more Pocket_feature_in_solid objects.

4.3.906 Point_direction to Cartesian_point

Each Point_direction refers to exactly one Cartesian_point in the role of location. Each Cartesian_point acts as location for zero or more Point_direction objects.

4.3.907 Point_direction_model to Point_direction_sequence

Each Point_direction_model refers to zero or more Point_direction_sequence objects in the role of point_direction. Each Point_direction_sequence acts as point_direction for zero or more Point_direction_model objects.

4.3.908 Point_direction_model to Point_sequence

Each Point_direction_model refers to zero or more Point_sequence objects in the role of point. Each Point_sequence acts as point for zero or more Point_direction_model objects.

4.3.909 Point_direction_sequence to Point_direction

Each Point_direction_sequence refers to zero or more Point_direction objects in the role of element. Each Point_direction acts as element for zero or more Point_direction_sequence objects.

NOTE – This assertion is established through point_direction_element_select.

4.3.910 Point_direction_sequence to Point_on_surface

Each Point_direction_sequence refers to zero or more Point_on_surface objects in the role of element. Each Point_on_surface acts as element for zero or more Point_direction_sequence objects.

NOTE – This assertion is established through point_direction_element_select.

4.3.911 Point_sequence to Cartesian_point

Each Point_sequence refers to one or more Cartesian_point objects in the role of element. Each Cartesian_point acts as element for zero or more Point_sequence objects.

4.3.912 Poly_loop to Cartesian_point

Each Poly_loop refers to three or more Cartesian_point objects in the role of polygon. Each Cartesian_point acts as polygon for zero or more Poly_loop objects.

4.3.913 Position_tolerance to Datum

Each Position_tolerance refers to one, two, or three Datum objects in the role of datum. Each Datum acts as datum for zero or more Position_tolerance objects.

4.3.914 Position_tolerance to Reference_plane

Each Position_tolerance refers to zero or one Reference_plane in the role of affected_plane. Each Reference_plane acts as affected_plane for zero or more Position_tolerance objects.

4.3.915 Process_cost_property to Property_value

Each Process_cost_property refers to one or more Property_value objects in the role of cost_property_value. Each Property_value acts as cost_property_value for zero or more Process_cost_property objects.

4.3.916 Process_operation to Cartesian_coordinate_space

Each Process_operation is_defined_in zero or one Cartesian_coordinate_space. Each Cartesian_coordinate_space is related to zero or more Process_operation objects.

4.3.917 Process_operation to Multi_language_string

Each Process_operation refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Process_operation objects.

NOTE – This assertion is established through string_select.

Each Process_operation refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Process_operation objects.

NOTE – This assertion is established through string_select.

4.3.918 Process_operation to Organization

Each Process_operation refers to zero or one Organization in the role of location. Each Organization acts as location for zero or more Process_operation objects.

4.3.919 Process_operation to Property_value

Each Process_operation refers to zero or one Property_value in the role of process_duration. Each Property_value acts as process_duration for zero or more Process_operation objects.

4.3.920 Process_operation_input_or_output to Assembly_component_relationship

Each Process_operation_input_or_output refers to zero or one Assembly_component_relationship in the role of element. Each Assembly_component_relationship acts as element for zero or more Process_operation_input_or_output objects.

NOTE – This assertion is established through process_operation_input_or_output_select.

4.3.921 Process_operation_input_or_output to Design_discipline_item_definition

Each Process_operation_input_or_output refers to zero or one Design_discipline_item_definition in the role of element. Each Design_discipline_item_definition acts as element for zero or more Process_operation_input_or_output objects.

NOTE – This assertion is established through process_operation_input_or_output_select.

4.3.922 Process_operation_input_or_output to Item_instance

Each Process_operation_input_or_output refers to zero or one Item_instance in the role of element. Each Item_instance acts as element for zero or more Process_operation_input_or_output objects.

NOTE – This assertion is established through process_operation_input_or_output_select.

4.3.923 Process_operation_input_or_output to Mated_item_association

Each Process_operation_input_or_output refers to zero or one Mated_item_association in the role of element. Each Mated_item_association acts as element for zero or more Process_operation_input_or_output objects.

NOTE – This assertion is established through process_operation_input_or_output_select.

4.3.924 Process_operation_input_or_output to Multi_language_string

Each Process_operation_input_or_output refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Process_operation_input_or_output objects.

NOTE – This assertion is established through string_select.

4.3.925 Process_operation_input_or_output to Process_plan_operation_assignment

Each Process_operation_input_or_output refers to exactly one Process_plan_operation_assignment in the role of operation. Each Process_plan_operation_assignment acts as operation for zero or more Process_operation_input_or_output objects.

4.3.926 Process_operation_input_or_output to Shape_aspect

Each Process_operation_input_or_output refers to zero or more Shape_aspect objects in the role of concerned_shape. Each Shape_aspect acts as concerned_shape for zero or more Process_operation_input_or_output objects.

4.3.927 Process_operation_input_or_output to Transformation

Each Process_operation_input_or_output refers to zero or one Transformation in the role of transformation. Each Transformation acts as transformation for zero or more Process_operation_input_or_output objects.

4.3.928 Process_operation_relationship to Multi_language_string

Each Process_operation_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Process_operation_relationship objects.

NOTE – This assertion is established through string_select.

4.3.929 Process_operation_relationship to Process_operation

Each Process_operation_relationship is related to exactly one Process_operation. Each Process_operation is referenced by zero or more Process_operation_relationship objects as related.

NOTE – The related Process_operation is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Process_operation_relationship is relating to exactly one Process_operation. Each Process_operation is referenced by zero or more Process_operation_relationship objects as relating.

NOTE – The relating Process_operation is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.930 Process_operation_relationship to Property_value

Each Process_operation_relationship refers to zero or one Property_value in the role of waiting_time. Each Property_value acts as waiting_time for zero or more Process_operation_relationship objects.

4.3.931 Process_operation_resource to Multi_language_string

Each Process_operation_resource refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Process_operation_resource objects.

NOTE – This assertion is established through string_select.

4.3.932 Process_operation_resource_assignment to Design_discipline_item_definition

Each Process_operation_resource_assignment refers to zero or one Design_discipline_item_definition in the role of resource. Each Design_discipline_item_definition acts as resource for zero or more Process_operation_resource_assignment objects.

NOTE – This assertion is established through resource_select.

4.3.933 Process_operation_resource_assignment to Multi_language_string

Each Process_operation_resource_assignment refers to zero or one Multi_language_string in the role of reason. Each Multi_language_string acts as reason for zero or more Process_operation_resource_assignment objects.

NOTE – This assertion is established through string_select.

4.3.934 Process_operation_resource_assignment to Process_operation

Each Process_operation_resource_assignment refers to exactly one Process_operation in the role of operation. Each Process_operation acts as operation for zero or more Process_operation_resource_assignment objects.

4.3.935 Process_operation_resource_assignment to Process_operation_resource

Each Process_operation_resource_assignment refers to zero or one Process_operation_resource in the role of resource. Each Process_operation_resource acts as resource for zero or more Process_-

operation_resource_assignment objects.

NOTE – This assertion is established through resource_select.

4.3.936 Process_operation_resource_assignment to Property_value

Each Process_operation_resource_assignment refers to zero or one Property_value in the role of process_duration_for_resource. Each Property_value acts as process_duration_for_resource for zero or more Process_operation_resource_assignment objects.

4.3.937 Process_operation_resource_assignment to Transformation

Each Process_operation_resource_assignment refers to zero or one Transformation in the role of transformation. Each Transformation acts as transformation for zero or more Process_operation_resource_assignment objects.

4.3.938 Process_plan to Item_version

Each Process_plan refers to zero or more Item_version objects in the role of produced_output. Each Item_version acts as produced_output for zero or more Process_plan objects.

4.3.939 Process_plan to Multi_language_string

Each Process_plan refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Process_plan objects.

NOTE – This assertion is established through string_select.

Each Process_plan refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Process_plan objects.

NOTE – This assertion is established through string_select.

4.3.940 Process_plan to Organization

Each Process_plan refers to zero or one Organization in the role of location. Each Organization acts as location for zero or more Process_plan objects.

4.3.941 Process_plan_operation_assignment to Process_operation

Each Process_plan_operation_assignment refers to exactly one Process_operation in the role of operation. Each Process_operation acts as operation for zero or more Process_plan_operation_assignment objects.

4.3.942 Process_plan_operation_assignment to Process_plan

Each Process_plan_operation_assignment refers to exactly one Process_plan in the role of plan. Each Process_plan acts as plan for zero or more Process_plan_operation_assignment objects.

4.3.943 Process_plan_relationship to Multi_language_string

Each Process_plan_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Process_plan_relationship objects.

NOTE – This assertion is established through string_select.

4.3.944 Process_plan_relationship to Process_plan

Each Process_plan_relationship is related to exactly one Process_plan. Each Process_plan is referenced by zero or more Process_plan_relationship objects as related.

NOTE – The related Process_plan is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Process_plan_relationship is relating to exactly one Process_plan. Each Process_plan is referenced by zero or more Process_plan_relationship objects as relating.

NOTE – The relating Process_plan is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.945 Process_property to Process_operation

Each Process_property refers to zero or one Process_operation in the role of described_object. Each Process_operation acts as described_object for zero or more Process_property objects.

NOTE – This assertion is established through process_information_select.

4.3.946 Process_property to Process_plan

Each Process_property refers to zero or one Process_plan in the role of described_object. Each Process_plan acts as described_object for zero or more Process_property objects.

NOTE – This assertion is established through process_information_select.

4.3.947 Process_quality_property to Multi_language_string

Each Process_quality_property refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Process_quality_property objects.

NOTE – This assertion is established through string_select.

4.3.948 Process_quality_property to Property_value

Each Process_quality_property refers to one or more Property_value objects in the role of quality_property_value. Each Property_value acts as quality_property_value for zero or more Process_quality_property objects.

4.3.949 Process_state to Design_discipline_item_definition

Each Process_state refers to exactly one Design_discipline_item_definition in the role of related_item_definition. Each Design_discipline_item_definition acts as related_item_definition for zero or more Process_state objects.

4.3.950 Product_class to Multi_language_string

Each Product_class refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Product_class objects.

NOTE – This assertion is established through string_select.

Each Product_class refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Product_class objects.

NOTE – This assertion is established through string_select.

4.3.951 Product_class to Product_component

Each Product_class refers to zero or one Product_component in the role of root. Each Product_component is_root_entry_for zero or one Product_class.

4.3.952 Product_class_hierarchy to Product_class

Each Product_class_hierarchy refers to exactly one Product_class in the role of sub_class. Each Product_class acts as sub_class for zero or more Product_class_hierarchy objects.

Each Product_class_hierarchy refers to exactly one Product_class in the role of super_class. Each Product_class acts as super_class for zero or more Product_class_hierarchy objects.

4.3.953 Product_component to Application_context

Each Product_component is_relevant_for one or more Application_context objects. Each Application_context is related to zero or more Product_component objects.

4.3.954 Product_component to Class_category_association

Each Product_component is_influenced_by zero or more Class_category_association objects. Each Class_category_association is related to zero or more Product_component objects.

4.3.955 Product_component to Multi_language_string

Each Product_component refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Product_component objects.

NOTE – This assertion is established through string_select.

Each Product_component refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Product_component objects.

NOTE – This assertion is established through string_select.

4.3.956 Product_component_relationship to Multi_language_string

Each Product_component_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Product_component_relationship objects.

NOTE – This assertion is established through string_select.

4.3.957 Product_component_relationship to Product_component

Each Product_component_relationship is related to exactly one Product_component. Each Product_component is referenced by zero or more Product_component_relationship objects as related.

NOTE – The related Product_component is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Product_component_relationship is relating to exactly one Product_component. Each Product_component is referenced by zero or more Product_component_relationship objects as relating.

NOTE – The relating Product_component is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.958 Product_function to Application_context

Each Product_function is_relevant_for one or more Application_context objects. Each Application_context is related to zero or more Product_function objects.

4.3.959 Product_function to Multi_language_string

Each Product_function refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Product_function objects.

NOTE – This assertion is established through string_select.

Each Product_function refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Product_function objects.

NOTE – This assertion is established through string_select.

4.3.960 Product_function_relationship to Multi_language_string

Each Product_function_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Product_function_relationship objects.

NOTE – This assertion is established through string_select.

4.3.961 Product_function_relationship to Product_function

Each Product_function_relationship is related to exactly one Product_function. Each Product_function is referenced by zero or more Product_function_relationship objects as related.

NOTE – The related Product_function is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Product_function_relationship is relating to exactly one Product_function. Each Product_function is referenced by zero or more Product_function_relationship objects as relating.

NOTE – The relating Product_function is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.962 Product_specification to Item_version

Each Product_specification refers to zero or one Item_version in the role of explicit_product_version. Each Item_version has implicit_product_version defined by zero or one Product_specification.

4.3.963 Product_specification to Multi_language_string

Each Product_specification refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Product_specification objects.

NOTE – This assertion is established through string_select.

4.3.964 Product_specification to Product_class

Each Product_specification refers to exactly one Product_class in the role of product_class. Each Product_class acts as product_class for zero or more Product_specification objects.

4.3.965 Product_specification to Specification

Each Product_specification refers to zero or more Specification objects in the role of specification. Each Specification acts as specification for zero or more Product_specification objects.

4.3.966 Profile to Axis_placement

Each Profile refers to zero or one Axis_placement in the role of parameter_reference. Each Axis_placement acts as parameter_reference for zero or more Profile objects.

4.3.967 Profile to Cartesian_coordinate_space

Each Profile is_defined_in exactly one Cartesian_coordinate_space. Each Cartesian_coordinate_space is related to zero or more Profile objects.

4.3.968 Project to Activity

Each Project refers to zero or more Activity objects in the role of work_program. Each Activity has associated_project defined by zero or one Project.

4.3.969 Project to Date_time

Each Project refers to zero or one Date_time in the role of actual_end_date. Each Date_time acts as actual_end_date for zero or more Project objects.

Each Project refers to zero or one Date_time in the role of actual_start_date. Each Date_time acts as actual_start_date for zero or more Project objects.

Each Project refers to zero or one Date_time in the role of planned_end_date. Each Date_time acts as planned_end_date for zero or more Project objects.

NOTE – This assertion is established through period_or_date_select.

Each Project refers to zero or one Date_time in the role of planned_start_date. Each Date_time acts as planned_start_date for zero or more Project objects.

NOTE – This assertion is established through event_or_date_select.

4.3.970 Project to Event_reference

Each Project refers to zero or one Event_reference in the role of planned_end_date. Each Event_reference acts as planned_end_date for zero or more Project objects.

NOTE – This assertion is established through period_or_date_select.

Each Project refers to zero or one Event_reference in the role of planned_start_date. Each Event_reference acts as planned_start_date for zero or more Project objects.

NOTE – This assertion is established through event_or_date_select.

4.3.971 Project to Multi_language_string

Each Project refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Project objects.

NOTE – This assertion is established through string_select.

Each Project refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Project objects.

NOTE – This assertion is established through string_select.

4.3.972 Project to Product_class

Each Project refers to zero or more Product_class objects in the role of affected_product_class. Each Product_class acts as affected_product_class for zero or more Project objects.

4.3.973 Project to Value_with_unit

Each Project refers to zero or one Value_with_unit in the role of planned_end_date. Each Value_with_unit acts as planned_end_date for zero or more Project objects.

NOTE – This assertion is established through period_or_date_select.

4.3.974 Project_relationship to Multi_language_string

Each Project_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Project_relationship objects.

NOTE – This assertion is established through string_select.

4.3.975 Project_relationship to Project

Each Project_relationship is related to exactly one Project. Each Project is referenced by zero or more Project_relationship objects as related.

NOTE – The related Project is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Project_relationship is relating to exactly one Project. Each Project is referenced by zero or more Project_relationship objects as relating.

NOTE – The relating Project is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.976 Projection to Shape_aspect

Each Projection refers to exactly one Shape_aspect in the role of projection_end. Each Shape_aspect acts as projection_end for zero or more Projection objects.

4.3.977 Projection to Tolerance_zone

Each Projection is_specifying exactly one Tolerance_zone. Each Tolerance_zone has created_by defined by zero, one, two, or three Projection objects.

4.3.978 Property_change to Change

Each Property_change is_describing exactly one Change. Each Change is related to zero or more Property_change objects.

4.3.979 Property_change to Multi_language_string

Each Property_change refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Property_change objects.

NOTE – This assertion is established through string_select.

4.3.980 Property_change to Property

Each Property_change refers to zero or more Property objects in the role of added_property. Each Property acts as added_property for zero or more Property_change objects.

Each Property_change refers to zero or more Property objects in the role of deleted_property. Each Property acts as deleted_property for zero or more Property_change objects.

4.3.981 Property_relationship to Multi_language_string

Each Property_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Property_relationship objects.

NOTE – This assertion is established through string_select.

4.3.982 Property_relationship to Property

Each Property_relationship is related to exactly one Property. Each Property is referenced by zero or more Property_relationship objects as related.

NOTE – The related Property is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Property_relationship is relating to exactly one Property. Each Property is referenced by zero or more Property_relationship objects as relating.

NOTE – The relating Property is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.983 Property_value to Multi_language_string

Each Property_value refers to zero or one Multi_language_string in the role of specified_value. Each Multi_language_string acts as specified_value for zero or more Property_value objects.

NOTE – This assertion is established through property_value_select and string_select.

4.3.984 Property_value to Value_with_unit

Each Property_value refers to zero or one Value_with_unit in the role of specified_value. Each Value_with_unit acts as specified_value for zero or more Property_value objects.

NOTE – This assertion is established through property_value_select.

4.3.985 Quantified_instance to Value_with_unit

Each Quantified_instance refers to exactly one Value_with_unit in the role of quantity. Each Value_with_unit acts as quantity for zero or more Quantified_instance objects.

4.3.986 Rectangular_boss to Rectangular_closed_profile

Each Rectangular_boss refers to exactly one Rectangular_closed_profile in the role of boundary. Each Rectangular_closed_profile acts as boundary for zero or more Rectangular_boss objects.

4.3.987 Rectangular_closed_pocket_in_solid to Rectangular_closed_profile

Each Rectangular_closed_pocket_in_solid refers to exactly one Rectangular_closed_profile in the role of boundary. Each Rectangular_closed_profile acts as boundary for zero or more Rectangular_closed_pocket_in_solid objects.

4.3.988 Rectangular_closed_profile to Feature_parameter

Each Rectangular_closed_profile refers to zero or one Feature_parameter in the role of corner_radius. Each Feature_parameter acts as corner_radius for zero or more Rectangular_closed_profile objects.

Each Rectangular_closed_profile refers to exactly one Feature_parameter in the role of profile_length. Each Feature_parameter acts as profile_length for zero or more Rectangular_closed_profile objects.

Each Rectangular_closed_profile refers to exactly one Feature_parameter in the role of profile_width. Each Feature_parameter acts as profile_width for zero or more Rectangular_closed_profile objects.

4.3.989 Rectangular_offset_pattern to Feature_parameter

Each Rectangular_offset_pattern refers to exactly one Feature_parameter in the role of offset_distance. Each Feature_parameter acts as offset_distance for zero or more Rectangular_offset_pattern objects.

4.3.990 Rectangular_open_pocket_in_solid to Square_u_profile

Each Rectangular_open_pocket_in_solid refers to exactly one Square_u_profile in the role of boundary. Each Square_u_profile acts as boundary for zero or more Rectangular_open_pocket_in_solid objects.

4.3.991 Rectangular_pattern to Feature_parameter

Each Rectangular_pattern refers to exactly one Feature_parameter in the role of column_spacing. Each Feature_parameter acts as column_spacing for zero or more Rectangular_pattern objects.

Each Rectangular_pattern refers to exactly one Feature_parameter in the role of row_spacing. Each Feature_parameter acts as row_spacing for zero or more Rectangular_pattern objects.

4.3.992 Rectangular_pattern to Rectangular_offset_pattern

Each Rectangular_pattern refers to zero or more Rectangular_offset_pattern objects in the role of relocated_feature. Each Rectangular_offset_pattern acts as relocated_feature for zero or more Rectangular_pattern objects.

4.3.993 Rectangular_pattern to Rectangular_omit_pattern

Each Rectangular_pattern refers to zero or more Rectangular_omit_pattern objects in the role of missing_feature. Each Rectangular_omit_pattern acts as missing_feature for zero or more Rectangular_pattern objects.

4.3.994 Recyclability to Multi_language_string

Each Recyclability refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Recyclability objects.

NOTE – This assertion is established through string_select.

4.3.995 Recyclability to Property_value

Each Recyclability refers to one or more Property_value objects in the role of recycling_class_value. Each Property_value acts as recycling_class_value for zero or more Recyclability objects.

4.3.996 Replaced_definition_relationship to Multi_language_string

Each Replaced_definition_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Replaced_definition_relationship objects.

NOTE – This assertion is established through string_select.

4.3.997 Replaced_usage_relationship to Multi_language_string

Each Replaced_usage_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Replaced_usage_relationship objects.

NOTE – This assertion is established through string_select.

4.3.998 Replicate_base_assignment to Feature_definition

Each Replicate_base_assignment refers to exactly one Feature_definition in the role of base_-feature. Each Feature_definition acts as base_feature for zero or more Replicate_base_assignment objects.

4.3.999 Replicate_base_assignment to Transformation

Each Replicate_base_assignment refers to zero or one Transformation in the role of transformation. Each Transformation acts as transformation for zero or more Replicate_base_assignment objects.

4.3.1000 Replicate_feature to Replicate_base_assignment

Each Replicate_feature refers to exactly one Replicate_base_assignment in the role of replicate_-base. Each Replicate_base_assignment acts as replicate_base for zero or more Replicate_feature objects.

4.3.1001 Retention_period to Activity

Each Retention_period is_applied_to zero or more Activity objects. Each Activity is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1002 Retention_period to Activity_element

Each Retention_period is_applied_to zero or more Activity_element objects. Each Activity_-element is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1003 Retention_period to Assembly_substitute_relationship

Each Retention_period is_applied_to zero or more Assembly_substitute_relationship objects. Each Assembly_substitute_relationship is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1004 Retention_period to Class_category_association

Each Retention_period is_applied_to zero or more Class_category_association objects. Each Class_category_association is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.1005 Retention_period to Class_condition_association

Each Retention_period is_applied_to zero or more Class_condition_association objects. Each Class_condition_association is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.1006 Retention_period to Class_function_association

Each Retention_period is_applied_to zero or more Class_function_association objects. Each Class_function_association is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.1007 Retention_period to Class_inclusion_association

Each Retention_period is_applied_to zero or more Class_inclusion_association objects. Each Class_inclusion_association is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.1008 Retention_period to Class_specification_association

Each Retention_period is_applied_to zero or more Class_specification_association objects. Each Class_specification_association is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.1009 Retention_period to Configuration

Each Retention_period is_applied_to zero or more Configuration objects. Each Configuration is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1010 Retention_period to Date_time

Each Retention_period refers to zero or one Date_time in the role of earliest_end_definition. Each Date_time acts as earliest_end_definition for zero or more Retention_period objects.

NOTE – This assertion is established through period_or_date_select.

Each Retention_period refers to zero or one Date_time in the role of latest_end_definition. Each Date_time acts as latest_end_definition for zero or more Retention_period objects.

NOTE – This assertion is established through period_or_date_select.

Each Retention_period refers to zero or one Date_time in the role of start_definition. Each Date_time acts as start_definition for zero or more Retention_period objects.

NOTE – This assertion is established through event_or_date_select.

4.3.1011 Retention_period to Design_constraint

Each Retention_period is_applied_to zero or more Design_constraint objects. Each Design_constraint is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.1012 Retention_period to Design_discipline_item_definition

Each Retention_period is_applied_to zero or more Design_discipline_item_definition objects. Each Design_discipline_item_definition is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and item_information_select.

4.3.1013 Retention_period to Drawing

Each Retention_period is_applied_to zero or more Drawing objects. Each Drawing is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1014 Retention_period to Event_reference

Each Retention_period refers to zero or one Event_reference in the role of earliest_end_definition. Each Event_reference acts as earliest_end_definition for zero or more Retention_period objects.

NOTE – This assertion is established through period_or_date_select.

Each Retention_period refers to zero or one Event_reference in the role of latest_end_definition. Each Event_reference acts as latest_end_definition for zero or more Retention_period objects.

NOTE – This assertion is established through period_or_date_select.

Each Retention_period refers to zero or one Event_reference in the role of start_definition. Each Event_reference acts as start_definition for zero or more Retention_period objects.

NOTE – This assertion is established through event_or_date_select.

4.3.1015 Retention_period to External_document

Each Retention_period is_applied_to zero or more External_document objects. Each External_document is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1016 Retention_period to Geometric_model

Each Retention_period is_applied_to zero or more Geometric_model objects. Each Geometric_model is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.1017 Retention_period to Item

Each Retention_period is_applied_to zero or more Item objects. Each Item is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1018 Retention_period to Item_definition_instance_relationship

Each Retention_period is_applied_to zero or more Item_definition_instance_relationship objects. Each Item_definition_instance_relationship is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_select, and item_relationship_select.

4.3.1019 Retention_period to Item_definition_relationship

Each Retention_period is_applied_to zero or more Item_definition_relationship objects. Each Item_definition_relationship is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_select, and item_relationship_select.

4.3.1020 Retention_period to Item_instance

Each Retention_period is_applied_to zero or more Item_instance objects. Each Item_instance is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and item_information_select.

4.3.1021 Retention_period to Item_instance_relationship

Each Retention_period is_applied_to zero or more Item_instance_relationship objects. Each Item_instance_relationship is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select, item_information_select, and item_relationship_select.

4.3.1022 Retention_period to Item_solution

Each Retention_period is_applied_to zero or more Item_solution objects. Each Item_solution is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.1023 Retention_period to Item_version

Each Retention_period is_applied_to zero or more Item_version objects. Each Item_version is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1024 Retention_period to Material

Each Retention_period is_applied_to zero or more Material objects. Each Material is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1025 Retention_period to Multi_language_string

Each Retention_period refers to zero or one Multi_language_string in the role of retention_purpose. Each Multi_language_string acts as retention_purpose for zero or more Retention_period objects.

NOTE – This assertion is established through string_select.

4.3.1026 Retention_period to Physical_instance

Each Retention_period is_applied_to zero or more Physical_instance objects. Each Physical_instance is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1027 Retention_period to Physical_instance_usage

Each Retention_period is_applied_to zero or more Physical_instance_usage objects. Each Physical_instance_usage is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1028 Retention_period to Process_plan

Each Retention_period is_applied_to zero or more Process_plan objects. Each Process_plan is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1029 Retention_period to Product_class

Each Retention_period is_applied_to zero or more Product_class objects. Each Product_class is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.1030 Retention_period to Product_component

Each Retention_period is_applied_to zero or more Product_component objects. Each Product_component is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.1031 Retention_period to Product_function

Each Retention_period is_applied_to zero or more Product_function objects. Each Product_function is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.1032 Retention_period to Product_specification

Each Retention_period is_applied_to zero or more Product_specification objects. Each Product_specification is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1033 Retention_period to Project

Each Retention_period is_applied_to zero or more Project objects. Each Project is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1034 Retention_period to Property

Each Retention_period is_applied_to zero or more Property objects. Each Property is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1035 Retention_period to Simulation_information

Each Retention_period is_applied_to zero or more Simulation_information objects. Each Simulation_information is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.1036 Retention_period to Specification

Each Retention_period is_applied_to zero or more Specification objects. Each Specification is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.1037 Retention_period to Specification_category

Each Retention_period is_applied_to zero or more Specification_category objects. Each Specification_category is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and specification_information_select.

4.3.1038 Retention_period to Specification_expression

Each Retention_period is_applied_to zero or more Specification_expression objects. Each Specification_expression is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.1039 Retention_period to Specification_inclusion

Each Retention_period is_applied_to zero or more Specification_inclusion objects. Each Specification_inclusion is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and configuration_information_select.

4.3.1040 Retention_period to Styled_model

Each Retention_period is_applied_to zero or more Styled_model objects. Each Styled_model is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select and model_select.

4.3.1041 Retention_period to Value_with_unit

Each Retention_period refers to zero or one Value_with_unit in the role of earliest_end_definition. Each Value_with_unit acts as earliest_end_definition for zero or more Retention_period objects.

NOTE – This assertion is established through period_or_date_select.

Each Retention_period refers to zero or one Value_with_unit in the role of latest_end_definition. Each Value_with_unit acts as latest_end_definition for zero or more Retention_period objects.

NOTE – This assertion is established through period_or_date_select.

4.3.1042 Retention_period to View_area

Each Retention_period is_applied_to zero or more View_area objects. Each View_area is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1043 Retention_period to Work_request

Each Retention_period is_applied_to zero or more Work_request objects. Each Work_request is related to zero or more Retention_period objects.

NOTE – This assertion is established through general_organizational_data_select.

4.3.1044 Rib_feature to Directed_taper

Each Rib_feature refers to zero or one Directed_taper in the role of draft_taper. Each Directed_taper acts as draft_taper for zero or more Rib_feature objects.

4.3.1045 Rib_feature to Feature_parameter

Each Rib_feature refers to zero or one Feature_parameter in the role of fillet_radius. Each Feature_parameter acts as fillet_radius for zero or more Rib_feature objects.

Each Rib_feature refers to zero or one Feature_parameter in the role of shoulder_radius. Each Feature_parameter acts as shoulder_radius for zero or more Rib_feature objects.

4.3.1046 Rib_feature to Open_profile

Each Rib_feature refers to exactly one Open_profile in the role of rib_profile. Each Open_profile acts as rib_profile for zero or more Rib_feature objects.

4.3.1047 Rib_feature to Path

Each Rib_feature refers to exactly one Path in the role of sweep_path. Each Path acts as sweep_path for zero or more Rib_feature objects.

4.3.1048 Rib_feature to Transformation

Each Rib_feature refers to zero or one Transformation in the role of profile_placement. Each Transformation acts as profile_placement for zero or more Rib_feature objects.

4.3.1049 Roughness to Curve_on_surface

Each Roughness refers to zero or one Curve_on_surface in the role of measuring_direction. Each Curve_on_surface acts as measuring_direction for zero or more Roughness objects.

4.3.1050 Roughness to Value_with_unit

Each Roughness refers to zero or one Value_with_unit in the role of cutoff_value. Each Value_with_unit acts as cutoff_value for zero or more Roughness objects.

Each Roughness refers to exactly one Value_with_unit in the role of roughness_max_value. Each Value_with_unit acts as roughness_max_value for zero or more Roughness objects.

Each Roughness refers to zero or one Value_with_unit in the role of roughness_min_value. Each Value_with_unit acts as roughness_min_value for zero or more Roughness objects.

Each Roughness refers to zero or one Value_with_unit in the role of sampling_length. Each Value_with_unit acts as sampling_length for zero or more Roughness objects.

4.3.1051 Round_hole to Angle_taper

Each Round_hole refers to zero or one Angle_taper in the role of taper. Each Angle_taper acts as taper for zero or more Round_hole objects.

NOTE – This assertion is established through taper_select.

4.3.1052 Round_hole to Chamfer_definition

Each Round_hole refers to zero or one Chamfer_definition in the role of top_chamfer. Each Chamfer_definition acts as top_chamfer for zero or more Round_hole objects.

4.3.1053 Round_hole to Circular_closed_profile

Each Round_hole refers to exactly one Circular_closed_profile in the role of boundary. Each Circular_closed_profile acts as boundary for zero or more Round_hole objects.

4.3.1054 Round_hole to Diameter_taper

Each Round_hole refers to zero or one Diameter_taper in the role of taper. Each Diameter_taper acts as taper for zero or more Round_hole objects.

NOTE – This assertion is established through taper_select.

4.3.1055 Round_hole to Directed_taper

Each Round_hole refers to zero or one Directed_taper in the role of taper. Each Directed_taper acts as taper for zero or more Round_hole objects.

NOTE – This assertion is established through taper_select.

4.3.1056 Round_hole to Hole_bottom_condition

Each Round_hole refers to exactly one Hole_bottom_condition in the role of bottom_condition. Each Hole_bottom_condition acts as bottom_condition for zero or more Round_hole objects.

4.3.1057 Round_hole to Linear_path

Each Round_hole refers to exactly one Linear_path in the role of sweep_path. Each Linear_path acts as sweep_path for zero or more Round_hole objects.

4.3.1058 Round_hole to Transformation

Each Round_hole refers to zero or one Transformation in the role of profile_placement. Each Transformation acts as profile_placement for zero or more Round_hole objects.

4.3.1059 Rounded_u_profile to Feature_parameter

Each Rounded_u_profile refers to exactly one Feature_parameter in the role of width. Each Feature_parameter acts as width for zero or more Rounded_u_profile objects.

4.3.1060 Same_time_machining_relationship to Geometric_model_relationship_with_transformation

Each Same_time_machining_relationship refers to zero or one Geometric_model_relationship_with_transformation in the role of transformation. Each Geometric_model_relationship_with_transformation acts as transformation for zero or more Same_time_machining_relationship objects.

NOTE – This assertion is established through transformation_select.

4.3.1061 Same_time_machining_relationship to Multi_language_string

Each Same_time_machining_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Same_time_machining_relationship objects.

NOTE – This assertion is established through string_select.

Each Same_time_machining_relationship refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Same_time_machining_relationship objects.

NOTE – This assertion is established through string_select.

4.3.1062 Same_time_machining_relationship to Template_instance

Each Same_time_machining_relationship refers to zero or one Template_instance in the role of transformation. Each Template_instance acts as transformation for zero or more Same_time_machining_relationship objects.

NOTE – This assertion is established through transformation_select.

4.3.1063 Security_classification to Activity

Each Security_classification is_applied_to zero or more Activity objects. Each Activity is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1064 Security_classification to Activity_element

Each Security_classification is_applied_to zero or more Activity_element objects. Each Activity_element is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1065 Security_classification to Assembly_substitute_relationship

Each Security_classification is_applied_to zero or more Assembly_substitute_relationship objects. Each Assembly_substitute_relationship is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1066 Security_classification to Class_function_association

Each Security_classification is_applied_to zero or more Class_function_association objects. Each Class_function_association is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select and specification_information_select.

4.3.1067 Security_classification to Configuration

Each Security_classification is_applied_to zero or more Configuration objects. Each Configuration is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1068 Security_classification to Design_constraint

Each Security_classification is_applied_to zero or more Design_constraint objects. Each Design_constraint is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select and specification_information_select.

4.3.1069 Security_classification to Drawing

Each Security_classification is_applied_to zero or more Drawing objects. Each Drawing is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1070 Security_classification to External_document

Each Security_classification is_applied_to zero or more External_document objects. Each External_document is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1071 Security_classification to Geometric_model

Each Security_classification is_applied_to zero or more Geometric_model objects. Each Geometric_model is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select and model_select.

4.3.1072 Security_classification to Item

Each Security_classification is_applied_to zero or more Item objects. Each Item is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1073 Security_classification to Item_solution

Each Security_classification is_applied_to zero or more Item_solution objects. Each Item_solution is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select and specification_information_select.

4.3.1074 Security_classification to Item_version

Each Security_classification is_applied_to zero or more Item_version objects. Each Item_version is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1075 Security_classification to Material

Each Security_classification is_applied_to zero or more Material objects. Each Material is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1076 Security_classification to Multi_language_string

Each Security_classification refers to zero or one Multi_language_string in the role of purpose. Each Multi_language_string acts as purpose for zero or more Security_classification objects.

NOTE – This assertion is established through string_select.

4.3.1077 Security_classification to Physical_instance

Each Security_classification is_applied_to zero or more Physical_instance objects. Each Physical_instance is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1078 Security_classification to Physical_instance_usage

Each Security_classification is_applied_to zero or more Physical_instance_usage objects. Each Physical_instance_usage is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1079 Security_classification to Process_plan

Each Security_classification is_applied_to zero or more Process_plan objects. Each Process_plan is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1080 Security_classification to Product_class

Each Security_classification is_applied_to zero or more Product_class objects. Each Product_class is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select and specification_information_select.

4.3.1081 Security_classification to Product_component

Each Security_classification is_applied_to zero or more Product_component objects. Each Product_component is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select and specification_information_select.

4.3.1082 Security_classification to Product_function

Each Security_classification is_applied_to zero or more Product_function objects. Each Product_function is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select and specification_information_select.

4.3.1083 Security_classification to Project

Each Security_classification is_applied_to zero or more Project objects. Each Project is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1084 Security_classification to Property

Each Security_classification is_applied_to zero or more Property objects. Each Property is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1085 Security_classification to Security_level

Each Security_classification refers to exactly one Security_level in the role of security_classification_level. Each Security_level acts as security_classification_level for zero or more Security_classification objects.

4.3.1086 Security_classification to Simulation_information

Each Security_classification is_applied_to zero or more Simulation_information objects. Each Simulation_information is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select and model_select.

4.3.1087 Security_classification to Specification

Each Security_classification is_applied_to zero or more Specification objects. Each Specification is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select and specification_information_select.

4.3.1088 Security_classification to Specification_category

Each Security_classification is_applied_to zero or more Specification_category objects. Each Specification_category is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select and specification_information_select.

4.3.1089 Security_classification to Styled_model

Each Security_classification is_applied_to zero or more Styled_model objects. Each Styled_model is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select and model_select.

4.3.1090 Security_classification to View_area

Each Security_classification is_applied_to zero or more View_area objects. Each View_area is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1091 Security_classification to Work_order

Each Security_classification is_applied_to zero or more Work_order objects. Each Work_order is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1092 Security_classification to Work_request

Each Security_classification is_applied_to zero or more Work_request objects. Each Work_request is related to zero or more Security_classification objects.

NOTE – This assertion is established through security_element_select.

4.3.1093 Security_level to Classification_system

Each Security_level refers to zero or one Classification_system in the role of used_classification_system. Each Classification_system acts as used_classification_system for zero or more Security_level objects.

4.3.1094 Selected_instance to Value_with_unit

Each Selected_instance refers to zero or one Value_with_unit in the role of maximum_quantity. Each Value_with_unit acts as maximum_quantity for zero or more Selected_instance objects.

Each Selected_instance refers to zero or one Value_with_unit in the role of minimum_quantity. Each Value_with_unit acts as minimum_quantity for zero or more Selected_instance objects.

4.3.1095 Shape_aspect to Item_shape

Each Shape_aspect refers to exactly one Item_shape in the role of composition. Each Item_shape acts as composition for zero or more Shape_aspect objects.

4.3.1096 Shape_aspect to Multi_language_string

Each Shape_aspect refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Shape_aspect objects.

NOTE – This assertion is established through string_select.

4.3.1097 Shape_aspect_relationship to Multi_language_string

Each Shape_aspect_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Shape_aspect_relationship objects.

NOTE – This assertion is established through string_select.

4.3.1098 Shape_aspect_relationship to Shape_aspect

Each Shape_aspect_relationship is related to exactly one Shape_aspect. Each Shape_aspect is referenced by zero or more Shape_aspect_relationship objects as related.

NOTE – The related Shape_aspect is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Shape_aspect_relationship is relating to exactly one Shape_aspect. Each Shape_aspect is referenced by zero or more Shape_aspect_relationship objects as relating.

NOTE – The relating Shape_aspect is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.1099 Shape_dependent_property to Item_shape

Each Shape_dependent_property refers to zero or one Item_shape in the role of described_object. Each Item_shape acts as described_object for zero or more Shape_dependent_property objects.

NOTE – This assertion is established through shape_select.

4.3.1100 Shape_dependent_property to Shape_aspect

Each Shape_dependent_property refers to zero or one Shape_aspect in the role of described_object. Each Shape_aspect acts as described_object for zero or more Shape_dependent_property objects.

NOTE – This assertion is established through shape_select.

4.3.1101 Shape_description_association to External_model

Each Shape_description_association refers to zero or one External_model in the role of defining_geometry. Each External_model acts as defining_geometry for zero or more Shape_description_association objects.

NOTE – This assertion is established through shape_definition_select.

4.3.1102 Shape_description_association to Feature_definition

Each Shape_description_association is_defining_shape_for zero or one Feature_definition. Each Feature_definition is related to zero or more Shape_description_association objects.

NOTE – This assertion is established through shape_information_select and feature_definition_select.

4.3.1103 Shape_description_association to General_feature

Each Shape_description_association is_defining_shape_for zero or one General_feature. Each General_feature is related to zero or more Shape_description_association objects.

NOTE – This assertion is established through shape_information_select and feature_definition_select.

4.3.1104 Shape_description_association to Geometric_model

Each Shape_description_association refers to zero or one Geometric_model in the role of defining_geometry. Each Geometric_model acts as defining_geometry for zero or more Shape_description_association objects.

NOTE – This assertion is established through shape_definition_select.

4.3.1105 Shape_description_association to Item_shape

Each Shape_description_association is_defining_shape_for zero or one Item_shape. Each Item_shape is related to zero or more Shape_description_association objects.

NOTE – This assertion is established through shape_information_select and shape_select.

4.3.1106 Shape_description_association to Shape_aspect

Each Shape_description_association is_defining_shape_for zero or one Shape_aspect. Each Shape_aspect is related to zero or more Shape_description_association objects.

NOTE – This assertion is established through shape_information_select and shape_select.

4.3.1107 Shape_description_association to Shape_aspect_relationship

Each Shape_description_association is_defining_shape_for zero or one Shape_aspect_relationship. Each Shape_aspect_relationship is related to zero or more Shape_description_association objects.

NOTE – This assertion is established through shape_information_select.

4.3.1108 Shape_description_association to Thread_feature

Each Shape_description_association is_defining_shape_for zero or one Thread_feature. Each Thread_feature is related to zero or more Shape_description_association objects.

NOTE – This assertion is established through shape_information_select and feature_definition_select.

4.3.1109 Shape_description_association to Transition_feature

Each Shape_description_association is_defining_shape_for zero or one Transition_feature. Each Transition_feature is related to zero or more Shape_description_association objects.

NOTE – This assertion is established through shape_information_select and feature_definition_select.

4.3.1110 Shell to Face

Each Shell refers to one or more Face objects in the role of face. Each Face acts as face for zero or more Shell objects.

4.3.1111 Simple_pair_range to Kinematic_pair

Each Simple_pair_range is_applied_to exactly one Kinematic_pair. Each Kinematic_pair is related to zero or more Simple_pair_range objects.

4.3.1112 Simulation_information to Design_discipline_item_definition

Each Simulation_information refers to zero or more Design_discipline_item_definition objects in the role of described_object. Each Design_discipline_item_definition acts as described_object for zero or more Simulation_information objects.

NOTE – This assertion is established through simulation_context_select and item_information_select.

4.3.1113 Simulation_information to Ground_representation

Each Simulation_information refers to exactly one Ground_representation in the role of ground. Each Ground_representation is_referenced_by one or more Simulation_information objects.

4.3.1114 Simulation_information to Item_definition_instance_relationship

Each Simulation_information refers to zero or more Item_definition_instance_relationship objects in the role of described_object. Each Item_definition_instance_relationship acts as described_object for zero or more Simulation_information objects.

NOTE – This assertion is established through simulation_context_select, item_information_select, and item_relationship_select.

4.3.1115 Simulation_information to Item_definition_relationship

Each Simulation_information refers to zero or more Item_definition_relationship objects in the role of described_object. Each Item_definition_relationship acts as described_object for zero or more Simulation_information objects.

NOTE – This assertion is established through simulation_context_select, item_information_select, and item_relationship_select.

4.3.1116 Simulation_information to Item_instance

Each Simulation_information refers to zero or more Item_instance objects in the role of described_object. Each Item_instance acts as described_object for zero or more Simulation_information objects.

NOTE – This assertion is established through simulation_context_select and item_information_select.

4.3.1117 Simulation_information to Item_instance_relationship

Each Simulation_information refers to zero or more Item_instance_relationship objects in the role of described_object. Each Item_instance_relationship acts as described_object for zero or more Simulation_information objects.

NOTE – This assertion is established through simulation_context_select, item_information_select, and item_relationship_select.

4.3.1118 Simulation_information to Item_solution

Each Simulation_information refers to zero or more Item_solution objects in the role of described_object. Each Item_solution acts as described_object for zero or more Simulation_information objects.

NOTE – This assertion is established through simulation_context_select.

4.3.1119 Simulation_information to Kinematic_mechanism

Each Simulation_information refers to one or more Kinematic_mechanism objects in the role of mechanism. Each Kinematic_mechanism is_referenced_by one or more Simulation_information objects.

4.3.1120 Simulation_information to Product_class

Each Simulation_information refers to zero or more Product_class objects in the role of described_object. Each Product_class acts as described_object for zero or more Simulation_information objects.

NOTE – This assertion is established through simulation_context_select.

4.3.1121 Simulation_information to Product_component

Each Simulation_information refers to zero or more Product_component objects in the role of described_object. Each Product_component acts as described_object for zero or more Simulation_information objects.

NOTE – This assertion is established through simulation_context_select.

4.3.1122 Simulation_information to Product_function

Each Simulation_information refers to zero or more Product_function objects in the role of described_object. Each Product_function acts as described_object for zero or more Simulation_information objects.

NOTE – This assertion is established through simulation_context_select.

4.3.1123 Simultaneous_activity to Activity

Each Simultaneous_activity refers to two or more Activity objects in the role of element. Each Activity acts as element for zero or more Simultaneous_activity objects.

4.3.1124 Simultaneous_activity to Multi_language_string

Each Simultaneous_activity refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Simultaneous_activity objects.

NOTE – This assertion is established through string_select.

4.3.1125 Simultaneous_process_operation to Process_operation

Each Simultaneous_process_operation refers to two or more Process_operation objects in the role of element. Each Process_operation acts as element for zero or more Simultaneous_process_operation objects.

4.3.1126 Size_dimension to Annotation_curve

Each Size_dimension is_applied_to zero or one Annotation_curve. Each Annotation_curve is related to zero or more Size_dimension objects.

NOTE – This assertion is established through element_with_dimension_select.

4.3.1127 Size_dimension to Constructive_geometric_element

Each Size_dimension is_applied_to zero or one Constructive_geometric_element. Each Constructive_geometric_element is related to zero or more Size_dimension objects.

NOTE – This assertion is established through element_with_dimension_select.

4.3.1128 Size_dimension to Shape_aspect

Each Size_dimension is_applied_to zero or one Shape_aspect. Each Shape_aspect is related to zero or more Size_dimension objects.

NOTE – This assertion is established through element_with_dimension_select.

4.3.1129 Slot_feature to Open_profile

Each Slot_feature refers to exactly one Open_profile in the role of slot_profile. Each Open_profile acts as slot_profile for zero or more Slot_feature objects.

4.3.1130 Slot_feature to Path

Each Slot_feature refers to exactly one Path in the role of sweep_path. Each Path acts as sweep_path for zero or more Slot_feature objects.

4.3.1131 Slot_feature to Slot_end_type

Each Slot_feature refers to exactly two Slot_end_type objects in the role of end_condition. Each Slot_end_type acts as end_condition for zero or more Slot_feature objects.

4.3.1132 Slot_feature to Transformation

Each Slot_feature refers to zero or one Transformation in the role of profile_placement. Each Transformation acts as profile_placement for zero or more Slot_feature objects.

4.3.1133 Solid_replica to B_rep_model

Each Solid_replica refers to zero or one B_rep_model in the role of replicated_solid. Each B_rep_model acts as replicated_solid for zero or more Solid_replica objects.

NOTE – This assertion is established through csg_select and non_primitive_solid_select.

4.3.1134 Solid_replica to Boolean_result

Each Solid_replica refers to zero or one Boolean_result in the role of replicated_solid. Each Boolean_result acts as replicated_solid for zero or more Solid_replica objects.

NOTE – This assertion is established through csg_select.

4.3.1135 Solid_replica to Csg_primitive

Each Solid_replica refers to zero or one Csg_primitive in the role of replicated_solid. Each Csg_primitive acts as replicated_solid for zero or more Solid_replica objects.

NOTE – This assertion is established through csg_select.

4.3.1136 Solid_replica to Faceted_b_rep_model

Each Solid_replica refers to zero or one Faceted_b_rep_model in the role of replicated_solid. Each Faceted_b_rep_model acts as replicated_solid for zero or more Solid_replica objects.

NOTE – This assertion is established through csg_select and non_primitive_solid_select.

4.3.1137 Solid_replica to Solid_replica

Each Solid_replica refers to zero or one Solid_replica in the role of replicated_solid. Each Solid_replica acts as replicated_solid for zero or more Solid_replica objects.

NOTE – This assertion is established through csg_select and non_primitive_solid_select.

4.3.1138 Solid_replica to Swept_face_solid

Each Solid_replica refers to zero or one Swept_face_solid in the role of replicated_solid. Each Swept_face_solid acts as replicated_solid for zero or more Solid_replica objects.

NOTE – This assertion is established through csg_select and non_primitive_solid_select.

4.3.1139 Solution_certification to Date_time

Each Solution_certification refers to exactly one Date_time in the role of certification_date. Each Date_time acts as certification_date for zero or more Solution_certification objects.

4.3.1140 Solution_certification to Item_solution

Each Solution_certification refers to exactly one Item_solution in the role of certified_solution. Each Item_solution acts as certified_solution for zero or more Solution_certification objects.

4.3.1141 Solution_certification to Multi_language_string

Each Solution_certification refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Solution_certification objects.

NOTE – This assertion is established through string_select.

Each Solution_certification refers to zero or one Multi_language_string in the role of purpose. Each Multi_language_string acts as purpose for zero or more Solution_certification objects.

NOTE – This assertion is established through string_select.

4.3.1142 Solution_instance_association to Item_instance

Each Solution_instance_association refers to exactly one Item_instance in the role of instance. Each Item_instance acts as instance for zero or more Solution_instance_association objects.

4.3.1143 Solution_instance_association to Item_solution

Each Solution_instance_association refers to exactly one Item_solution in the role of solution. Each Item_solution acts as solution for zero or more Solution_instance_association objects.

4.3.1144 Specification to Multi_language_string

Each Specification refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Specification objects.

NOTE – This assertion is established through string_select.

Each Specification refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Specification objects.

NOTE – This assertion is established through string_select.

4.3.1145 Specification to Specification_category

Each Specification refers to exactly one Specification_category in the role of category. Each Specification_category acts as category for zero or more Specification objects.

4.3.1146 Specification_category to Multi_language_string

Each Specification_category refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Specification_category objects.

NOTE – This assertion is established through string_select.

4.3.1147 Specification_expression to Multi_language_string

Each Specification_expression refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Specification_expression objects.

NOTE – This assertion is established through string_select.

4.3.1148 Specification_expression to Specification

Each Specification_expression refers to zero or more Specification objects in the role of operand. Each Specification acts as operand for zero or more Specification_expression objects.

NOTE – This assertion is established through specification_operand_select.

4.3.1149 Specification_expression to Specification_expression

Each Specification_expression refers to zero or more Specification_expression objects in the role of operand. Each Specification_expression acts as operand for zero or more Specification_expression objects.

NOTE – This assertion is established through specification_operand_select.

4.3.1150 Specification_inclusion to Multi_language_string

Each Specification_inclusion refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Specification_inclusion objects.

NOTE – This assertion is established through string_select.

4.3.1151 Specification_inclusion to Specification

Each Specification_inclusion refers to zero or one Specification in the role of if_condition. Each Specification acts as if_condition for zero or more Specification_inclusion objects.

NOTE – This assertion is established through specification_operand_select.

Each Specification_inclusion refers to zero or one Specification in the role of included_specification. Each Specification acts as included_specification for zero or more Specification_inclusion objects.

NOTE – This assertion is established through specification_operand_select.

4.3.1152 Specification_inclusion to Specification_expression

Each Specification_inclusion refers to zero or one Specification_expression in the role of if_condition. Each Specification_expression acts as if_condition for zero or more Specification_inclusion objects.

NOTE – This assertion is established through specification_operand_select.

Each Specification_inclusion refers to zero or one Specification_expression in the role of included_specification. Each Specification_expression acts as included_specification for zero or more Specification_inclusion objects.

NOTE – This assertion is established through specification_operand_select.

4.3.1153 Specified_higher_usage to Assembly_component_relationship

Each Specified_higher_usage refers to exactly one Assembly_component_relationship in the role of upper_usage. Each Assembly_component_relationship acts as upper_usage for zero or more Specified_higher_usage objects.

4.3.1154 Specified_higher_usage to Next_higher_assembly

Each Specified_higher_usage refers to exactly one Next_higher_assembly in the role of next_usage. Each Next_higher_assembly acts as next_usage for zero or more Specified_higher_usage objects.

4.3.1155 Spherical_bottom to Feature_parameter

Each Spherical_bottom refers to exactly one Feature_parameter in the role of radius. Each Feature_parameter acts as radius for zero or more Spherical_bottom objects.

4.3.1156 Square_u_profile to Feature_parameter

Each Square_u_profile refers to exactly one Feature_parameter in the role of first_angle. Each Feature_parameter acts as first_angle for zero or more Square_u_profile objects.

Each Square_u_profile refers to zero or one Feature_parameter in the role of first_radius. Each Feature_parameter acts as first_radius for zero or more Square_u_profile objects.

Each Square_u_profile refers to exactly one Feature_parameter in the role of second_angle. Each Feature_parameter acts as second_angle for zero or more Square_u_profile objects.

Each Square_u_profile refers to zero or one Feature_parameter in the role of second_radius. Each Feature_parameter acts as second_radius for zero or more Square_u_profile objects.

Each Square_u_profile refers to exactly one Feature_parameter in the role of width. Each Feature_parameter acts as width for zero or more Square_u_profile objects.

4.3.1157 Straightness_tolerance to Reference_plane

Each Straightness_tolerance refers to zero or one Reference_plane in the role of affected_plane. Each Reference_plane acts as affected_plane for zero or more Straightness_tolerance objects.

4.3.1158 Structuring_method_hierarchy to Group

Each Structuring_method_hierarchy is related to exactly one Group. Each Group is referenced by zero or more Structuring_method_hierarchy objects as related.

NOTE – The related Group is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Structuring_method_hierarchy is relating to zero or one Group. Each Group is referenced by zero or more Structuring_method_hierarchy objects as relating.

NOTE – The relating Group is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

NOTE – This assertion is established through structuring_method_select.

4.3.1159 Structuring_method_hierarchy to Layer

Each Structuring_method_hierarchy is relating to zero or one Layer. Each Layer is referenced by zero or more Structuring_method_hierarchy objects as relating.

NOTE – The relating Layer is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

NOTE – This assertion is established through structuring_method_select.

4.3.1160 Styled_element to Appearance_assignment

Each Styled_element refers to exactly one Appearance_assignment in the role of appearance. Each Appearance_assignment acts as appearance for zero or more Styled_element objects.

4.3.1161 Styled_geometric_model to Geometric_model

Each Styled_geometric_model refers to exactly one Geometric_model in the role of styled_geometry. Each Geometric_model acts as styled_geometry for zero or more Styled_geometric_model objects.

4.3.1162 Styled_model to Multi_language_string

Each Styled_model refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Styled_model objects.

NOTE – This assertion is established through string_select.

4.3.1163 Styled_model to Styled_element

Each Styled_model refers to one or more Styled_element objects in the role of element. Each Styled_element acts as element for zero or more Styled_model objects.

4.3.1164 Styled_model to Template_instance

Each Styled_model refers to zero or more Template_instance objects in the role of additional_elements. Each Template_instance acts as additional_elements for zero or more Styled_model objects.

4.3.1165 Styled_model_element to Detailed_model_element

Each Styled_model_element refers to exactly one Detailed_model_element in the role of element. Each Detailed_model_element acts as element for zero or more Styled_model_element objects.

4.3.1166 Styled_model_relationship to Multi_language_string

Each Styled_model_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Styled_model_relationship objects.

NOTE – This assertion is established through string_select.

4.3.1167 Styled_model_relationship to Styled_model

Each Styled_model_relationship is related to exactly one Styled_model. Each Styled_model is referenced by zero or more Styled_model_relationship objects as related.

NOTE – The related Styled_model is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each Styled_model_relationship is relating to exactly one Styled_model. Each Styled_model is referenced by zero or more Styled_model_relationship objects as relating.

NOTE – The relating Styled_model is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.1168 Surface_appearance to Surface_appearance_element

Each Surface_appearance refers to one or more Surface_appearance_element objects in the role of element. Each Surface_appearance_element acts as element for zero or more Surface_appearance objects.

4.3.1169 Surface_appearance_element to Visual_appearance

Each Surface_appearance_element refers to zero or one Visual_appearance in the role of surface_condition. Each Visual_appearance acts as surface_condition for zero or more Surface_appearance_element objects.

4.3.1170 Surface_condition to Item_shape

Each Surface_condition is_applied_to zero or more Item_shape objects. Each Item_shape is related to zero or more Surface_condition objects.

NOTE – This assertion is established through shape_select.

4.3.1171 Surface_condition to Process_operation

Each Surface_condition is_caused_by zero or one Process_operation. Each Process_operation is related to zero or more Surface_condition objects.

4.3.1172 Surface_condition to Shape_aspect

Each Surface_condition is_applied_to zero or more Shape_aspect objects. Each Shape_aspect is related to zero or more Surface_condition objects.

NOTE – This assertion is established through shape_select.

4.3.1173 Surface_model to Shell

Each Surface_model refers to zero or more Shell objects in the role of element. Each Shell acts as element for zero or more Surface_model objects.

4.3.1174 Surface_profile_tolerance to Datum

Each Surface_profile_tolerance refers to zero, one, two, or three Datum objects in the role of datum. Each Datum acts as datum for zero or more Surface_profile_tolerance objects.

4.3.1175 Swept_face_solid to Face

Each Swept_face_solid refers to exactly one Face in the role of face. Each Face acts as face for zero or more Swept_face_solid objects.

4.3.1176 Symmetry_tolerance to Datum

Each Symmetry_tolerance refers to exactly one Datum in the role of datum. Each Datum acts as datum for zero or more Symmetry_tolerance objects.

4.3.1177 Symmetry_tolerance to Reference_plane

Each Symmetry_tolerance refers to zero or one Reference_plane in the role of affected_plane. Each Reference_plane acts as affected_plane for zero or more Symmetry_tolerance objects.

4.3.1178 Tactile_appearance to Multi_language_string

Each Tactile_appearance refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Tactile_appearance objects.

NOTE – This assertion is established through string_select.

Each Tactile_appearance refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Tactile_appearance objects.

NOTE – This assertion is established through string_select.

4.3.1179 Tactile_appearance to Value_with_unit

Each Tactile_appearance refers to zero or one Value_with_unit in the role of depth. Each Value_with_unit acts as depth for zero or more Tactile_appearance objects.

4.3.1180 Target_area to Shape_aspect

Each Target_area is_defined_by exactly one Shape_aspect. Each Shape_aspect is related to zero or more Target_area objects.

4.3.1181 Technical_solution to Multi_language_string

Each Technical_solution refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Technical_solution objects.

NOTE – This assertion is established through string_select.

4.3.1182 Tee_profile to Feature_parameter

Each Tee_profile refers to exactly one Feature_parameter in the role of cross_bar_depth. Each Feature_parameter acts as cross_bar_depth for zero or more Tee_profile objects.

Each Tee_profile refers to exactly one Feature_parameter in the role of cross_bar_width. Each Feature_parameter acts as cross_bar_width for zero or more Tee_profile objects.

Each Tee_profile refers to exactly one Feature_parameter in the role of depth. Each Feature_parameter acts as depth for zero or more Tee_profile objects.

Each Tee_profile refers to exactly one Feature_parameter in the role of first_angle. Each Feature_parameter acts as first_angle for zero or more Tee_profile objects.

Each Tee_profile refers to exactly one Feature_parameter in the role of first_offset. Each Feature_parameter acts as first_offset for zero or more Tee_profile objects.

Each Tee_profile refers to zero or one Feature_parameter in the role of radius. Each Feature_parameter acts as radius for zero or more Tee_profile objects.

Each Tee_profile refers to exactly one Feature_parameter in the role of second_angle. Each Feature_parameter acts as second_angle for zero or more Tee_profile objects.

Each Tee_profile refers to exactly one Feature_parameter in the role of second_offset. Each Feature_parameter acts as second_offset for zero or more Tee_profile objects.

Each Tee_profile refers to exactly one Feature_parameter in the role of width. Each Feature_parameter acts as width for zero or more Tee_profile objects.

4.3.1183 Template_instance to External_model

Each Template_instance refers to zero or one External_model in the role of template_definition. Each External_model acts as template_definition for zero or more Template_instance objects.

NOTE – This assertion is established through template_definition_select.

4.3.1184 Template_instance to Geometric_model

Each Template_instance refers to zero or one Geometric_model in the role of template_definition. Each Geometric_model acts as template_definition for zero or more Template_instance objects.

NOTE – This assertion is established through template_definition_select.

4.3.1185 Template_instance to Styled_model

Each Template_instance refers to zero or one Styled_model in the role of template_definition. Each Styled_model acts as template_definition for zero or more Template_instance objects.

NOTE – This assertion is established through template_definition_select.

4.3.1186 Template_instance to Transformation

Each Template_instance refers to exactly one Transformation in the role of transformation. Each Transformation acts as transformation for zero or more Template_instance objects.

4.3.1187 Thickness to Material_side_specification

Each Thickness refers to zero or one Material_side_specification in the role of material_side. Each Material_side_specification acts as material_side for zero or more Thickness objects.

4.3.1188 Thickness to Multi_language_string

Each Thickness refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Thickness objects.

NOTE – This assertion is established through string_select.

4.3.1189 Thickness to Property_value

Each Thickness refers to one or more Property_value objects in the role of thickness_value. Each Property_value acts as thickness_value for zero or more Thickness objects.

4.3.1190 Thread_feature to Axis_placement

Each Thread_feature refers to exactly one Axis_placement in the role of orientation. Each Axis_placement acts as orientation for zero or more Thread_feature objects.

4.3.1191 Thread_feature to Thread_specification

Each Thread_feature refers to exactly one Thread_specification in the role of thread_definition. Each Thread_specification acts as thread_definition for zero or more Thread_feature objects.

4.3.1192 Tolerance_table_cell to General_tolerance_table

Each Tolerance_table_cell refers to exactly one General_tolerance_table in the role of table. Each General_tolerance_table has cell defined by one or more Tolerance_table_cell objects.

4.3.1193 Tolerance_table_cell to Multi_language_string

Each Tolerance_table_cell refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Tolerance_table_cell objects.

NOTE – This assertion is established through string_select.

4.3.1194 Tolerance_table_cell to Plus_minus_toleranced_datum

Each Tolerance_table_cell refers to zero or one Plus_minus_toleranced_datum in the role of cell_value. Each Plus_minus_toleranced_datum acts as cell_value for zero or more Tolerance_table_cell objects.

NOTE – This assertion is established through cell_entry_select.

4.3.1195 Tolerance_table_cell to Significant_number_of_digits

Each Tolerance_table_cell refers to zero or one Significant_number_of_digits in the role of limit. Each Significant_number_of_digits acts as limit for zero or more Tolerance_table_cell objects.

NOTE – This assertion is established through limit_select.

4.3.1196 Tolerance_table_cell to Upper_lower_limit

Each Tolerance_table_cell refers to zero or one Upper_lower_limit in the role of limit. Each Upper_lower_limit acts as limit for zero or more Tolerance_table_cell objects.

NOTE – This assertion is established through limit_select.

4.3.1197 Tolerance_table_cell to Upper_lower_toleranced_datum

Each Tolerance_table_cell refers to zero or one Upper_lower_toleranced_datum in the role of cell_value. Each Upper_lower_toleranced_datum acts as cell_value for zero or more Tolerance_table_cell objects.

NOTE – This assertion is established through cell_entry_select.

4.3.1198 Tolerance_zone to Geometric_tolerance

Each Tolerance_zone is_zone_for one or more Geometric_tolerance objects. Each Geometric_tolerance is related to zero or more Tolerance_zone objects.

4.3.1199 Tolerance_zone_definition to Shape_aspect

Each Tolerance_zone_definition refers to exactly one Shape_aspect in the role of first_element. Each Shape_aspect acts as first_element for zero or more Tolerance_zone_definition objects.

Each Tolerance_zone_definition refers to zero or one Shape_aspect in the role of second_element. Each Shape_aspect acts as second_element for zero or more Tolerance_zone_definition objects.

4.3.1200 Tolerance_zone_definition to Tolerance_zone

Each Tolerance_zone_definition is_defining exactly one Tolerance_zone. Each Tolerance_zone is related to zero or more Tolerance_zone_definition objects.

4.3.1201 Tool_part_relationship to Geometric_model_relationship_with_transformation

Each Tool_part_relationship refers to zero or one Geometric_model_relationship_with_transformation in the role of transformation. Each Geometric_model_relationship_with_transformation acts as transformation for zero or more Tool_part_relationship objects.

NOTE – This assertion is established through transformation_select.

4.3.1202 Tool_part_relationship to Multi_language_string

Each Tool_part_relationship refers to zero or one Multi_language_string in the role of used_technology_description. Each Multi_language_string acts as used_technology_description for zero or more Tool_part_relationship objects.

NOTE – This assertion is established through string_select.

4.3.1203 Tool_part_relationship to Template_instance

Each Tool_part_relationship refers to zero or one Template_instance in the role of transformation. Each Template_instance acts as transformation for zero or more Tool_part_relationship objects.

NOTE – This assertion is established through transformation_select.

4.3.1204 Topologically_connected_set to Face

Each Topologically_connected_set refers to zero or more Face objects in the role of element. Each Face acts as element for zero or more Topologically_connected_set objects.

NOTE – This assertion is established through connected_element_select.

4.3.1205 Topologically_connected_set to Oriented_edge_curve

Each Topologically_connected_set refers to zero or more Oriented_edge_curve objects in the role of element. Each Oriented_edge_curve acts as element for zero or more Topologically_connected_set objects.

NOTE – This assertion is established through connected_element_select.

4.3.1206 Total_runout_tolerance to Datum

Each Total_runout_tolerance refers to one or two Datum objects in the role of datum. Each Datum acts as datum for zero or more Total_runout_tolerance objects.

4.3.1207 Transition_feature to Face

Each Transition_feature refers to exactly one Face in the role of face_1. Each Face acts as face_1 for zero or more Transition_feature objects.

Each Transition_feature refers to exactly one Face in the role of face_2. Each Face acts as face_2 for zero or more Transition_feature objects.

4.3.1208 Treatment_result to Multi_language_string

Each Treatment_result refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Treatment_result objects.

NOTE – This assertion is established through string_select.

Each Treatment_result refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Treatment_result objects.

NOTE – This assertion is established through string_select.

Each Treatment_result refers to zero or one Multi_language_string in the role of purpose. Each Multi_language_string acts as purpose for zero or more Treatment_result objects.

NOTE – This assertion is established through string_select.

4.3.1209 Value_termination to Feature_parameter

Each Value_termination refers to exactly one Feature_parameter in the role of angle. Each Feature_parameter acts as angle for zero or more Value_termination objects.

Each Value_termination refers to exactly one Feature_parameter in the role of depth. Each Feature_parameter acts as depth for zero or more Value_termination objects.

4.3.1210 Vee_profile to Feature_parameter

Each Vee_profile refers to exactly one Feature_parameter in the role of profile_angle. Each Feature_parameter acts as profile_angle for zero or more Vee_profile objects.

Each Vee_profile refers to zero or one Feature_parameter in the role of radius. Each Feature_parameter acts as radius for zero or more Vee_profile objects.

Each Vee_profile refers to exactly one Feature_parameter in the role of tilt_angle. Each Feature_parameter acts as tilt_angle for zero or more Vee_profile objects.

4.3.1211 Vertex_loop to Vertex_point

Each Vertex_loop refers to exactly one Vertex_point in the role of loop_vertex. Each Vertex_point acts as loop_vertex for zero or more Vertex_loop objects.

4.3.1212 Vertex_point to Point

Each Vertex_point refers to exactly one Point in the role of vertex_geometry. Each Point acts as vertex_geometry for zero or more Vertex_point objects.

4.3.1213 View to Cartesian_coordinate_space_2d

Each View refers to exactly one Cartesian_coordinate_space_2d in the role of view_space. Each Cartesian_coordinate_space_2d acts as view_space for zero or more View objects.

4.3.1214 View to Drawing_sheet

Each View refers to zero or one Drawing_sheet in the role of layout. Each Drawing_sheet acts as layout for zero or more View objects.

NOTE – This assertion is established through area_select.

4.3.1215 View to Multi_language_string

Each View refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more View objects.

NOTE – This assertion is established through string_select.

Each View refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more View objects.

NOTE – This assertion is established through string_select.

4.3.1216 View to Transformation_2d

Each View refers to exactly one Transformation_2d in the role of transformation. Each Transformation_2d acts as transformation for zero or more View objects.

4.3.1217 View to View_area

Each View refers to zero or one View_area in the role of layout. Each View_area acts as layout for zero or more View objects.

NOTE – This assertion is established through area_select.

4.3.1218 View_area to Cartesian_coordinate_space_2d

Each View_area refers to exactly one Cartesian_coordinate_space_2d in the role of area_space. Each Cartesian_coordinate_space_2d acts as area_space for zero or more View_area objects.

4.3.1219 View_area to Multi_language_string

Each View_area refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more View_area objects.

NOTE – This assertion is established through string_select.

Each View_area refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more View_area objects.

NOTE – This assertion is established through string_select.

4.3.1220 View_area_assignment to Class_function_association

Each View_area_assignment is_describing zero or one Class_function_association. Each Class_function_association is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.1221 View_area_assignment to Design_constraint

Each View_area_assignment is_describing zero or one Design_constraint. Each Design_constraint is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.1222 View_area_assignment to Design_discipline_item_definition

Each View_area_assignment is_describing zero or one Design_discipline_item_definition. Each Design_discipline_item_definition is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select and item_information_select.

4.3.1223 View_area_assignment to Item_definition_instance_relationship

Each View_area_assignment is_describing zero or one Item_definition_instance_relationship. Each Item_definition_instance_relationship is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select, item_information_select, and item_relationship_select.

4.3.1224 View_area_assignment to Item_definition_relationship

Each View_area_assignment is_describing zero or one Item_definition_relationship. Each Item_definition_relationship is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select, item_information_select, and item_relationship_select.

4.3.1225 View_area_assignment to Item_instance

Each View_area_assignment is_describing zero or one Item_instance. Each Item_instance is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select and item_information_select.

4.3.1226 View_area_assignment to Item_instance_relationship

Each View_area_assignment is_describing zero or one Item_instance_relationship. Each Item_instance_relationship is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select, item_information_select, and item_relationship_select.

4.3.1227 View_area_assignment to Item_solution

Each View_area_assignment is_describing zero or one Item_solution. Each Item_solution is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.1228 View_area_assignment to Process_operation

Each View_area_assignment is_describing zero or one Process_operation. Each Process_operation is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select and process_information_select.

4.3.1229 View_area_assignment to Process_plan

Each View_area_assignment is_describing zero or one Process_plan. Each Process_plan is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select and process_information_select.

4.3.1230 View_area_assignment to Product_class

Each View_area_assignment is_describing zero or one Product_class. Each Product_class is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.1231 View_area_assignment to Product_component

Each View_area_assignment is_describing zero or one Product_component. Each Product_component is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.1232 View_area_assignment to Product_function

Each View_area_assignment is_describing zero or one Product_function. Each Product_function is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.1233 View_area_assignment to Specification

Each View_area_assignment is_describing zero or one Specification. Each Specification is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.1234 View_area_assignment to Specification_category

Each View_area_assignment is_describing zero or one Specification_category. Each Specification_category is related to zero or more View_area_assignment objects.

NOTE – This assertion is established through general_select and specification_information_select.

4.3.1235 View_area_assignment to View_area

Each View_area_assignment refers to exactly one View_area in the role of view_area. Each View_area acts as view_area for zero or more View_area_assignment objects.

4.3.1236 View_area_relationship to Multi_language_string

Each View_area_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more View_area_relationship objects.

NOTE – This assertion is established through string_select.

4.3.1237 View_area_relationship to View_area

Each View_area_relationship is related to exactly one View_area. Each View_area is referenced by zero or more View_area_relationship objects as related.

NOTE – The related View_area is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each View_area_relationship is relating to exactly one View_area. Each View_area is referenced by zero or more View_area_relationship objects as relating.

NOTE – The relating View_area is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.1238 View_placed_annotation to Annotation_element

Each View_placed_annotation refers to exactly one Annotation_element in the role of element. Each Annotation_element acts as element for zero or more View_placed_annotation objects.

4.3.1239 View_placed_annotation to Appearance_assignment

Each View_placed_annotation refers to exactly one Appearance_assignment in the role of appearance. Each Appearance_assignment acts as appearance for zero or more View_placed_annotation objects.

4.3.1240 View_placed_annotation to View

Each View_placed_annotation refers to exactly one View in the role of reference_view. Each View acts as reference_view for zero or more View_placed_annotation objects.

4.3.1241 View_relationship to Multi_language_string

Each View_relationship refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more View_relationship objects.

NOTE – This assertion is established through string_select.

4.3.1242 View_relationship to View

Each View_relationship is related to exactly one View. Each View is referenced by zero or more View_relationship objects as related.

NOTE – The related View is the secondary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the child object.

Each View_relationship is relating to exactly one View. Each View is referenced by zero or more View_relationship objects as relating.

NOTE – The relating View is the primary object in the relationship, i.e., in a hierarchical relationship structure it plays the role of the parent object.

4.3.1243 Visual_appearance to Multi_language_string

Each Visual_appearance refers to zero or one Multi_language_string in the role of colour_name. Each Multi_language_string acts as colour_name for zero or more Visual_appearance objects.

NOTE – This assertion is established through string_select.

Each Visual_appearance refers to zero or one Multi_language_string in the role of name. Each Multi_language_string acts as name for zero or more Visual_appearance objects.

NOTE – This assertion is established through string_select.

4.3.1244 Wireframe_model_2d to Curve_set_2d

Each Wireframe_model_2d refers to zero or more Curve_set_2d objects in the role of element. Each Curve_set_2d acts as element for zero or more Wireframe_model_2d objects.

4.3.1245 Wireframe_model_3d to Curve_set_3d

Each Wireframe_model_3d refers to zero or more Curve_set_3d objects in the role of element. Each Curve_set_3d acts as element for zero or more Wireframe_model_3d objects.

4.3.1246 Woodruff_slot_end_type to Feature_parameter

Each Woodruff_slot_end_type refers to exactly one Feature_parameter in the role of radius. Each Feature_parameter acts as radius for zero or more Woodruff_slot_end_type objects.

4.3.1247 Work_order to Activity

Each Work_order is_controlling one or more Activity objects. Each Activity has authorization defined by zero or one Work_order.

4.3.1248 Work_order to Multi_language_string

Each Work_order refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Work_order objects.

NOTE – This assertion is established through string_select.

4.3.1249 Work_request to Activity_method

Each Work_request refers to zero or more Activity_method objects in the role of recommended_-method. Each Activity_method acts as recommended_method for zero or more Work_request objects.

The assertion between work_request and activity_method in the role of scope exists only because of the select type structure and shall not be instantiated.

NOTE – This assertion is established through activity_element_select.

4.3.1250 Work_request to Assembly_substitute_relationship

Each Work_request refers to zero or more Assembly_substitute_relationship objects in the role of scope. Each Assembly_substitute_relationship acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_-effectivity_select.

4.3.1251 Work_request to Class_category_association

Each Work_request refers to zero or more Class_category_association objects in the role of scope. Each Class_category_association acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and configuration_information_select.

4.3.1252 Work_request to Class_condition_association

Each Work_request refers to zero or more Class_condition_association objects in the role of scope. Each Class_condition_association acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and configuration_information_select.

4.3.1253 Work_request to Class_function_association

Each Work_request refers to zero or more Class_function_association objects in the role of scope. Each Class_function_association acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.1254 Work_request to Class_inclusion_association

Each Work_request refers to zero or more Class_inclusion_association objects in the role of scope. Each Class_inclusion_association acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and configuration_information_select.

4.3.1255 Work_request to Class_specification_association

Each Work_request refers to zero or more Class_specification_association objects in the role of scope. Each Class_specification_association acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and configuration_information_select.

4.3.1256 Work_request to Configuration

Each Work_request refers to zero or more Configuration objects in the role of scope. Each Configuration acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select.

4.3.1257 Work_request to Date_and_person_or_organization

Each Work_request refers to one or more Date_and_person_or_organization objects in the role of notified_person. Each Date_and_person_or_organization acts as notified_person for zero or more Work_request objects.

Each Work_request refers to zero or one Date_and_person_or_organization in the role of requestor. Each Date_and_person_or_organization acts as requestor for zero or more Work_request objects.

4.3.1258 Work_request to Design_constraint

Each Work_request refers to zero or more Design_constraint objects in the role of scope. Each Design_constraint acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.1259 Work_request to Drawing

Each Work_request refers to zero or more Drawing objects in the role of scope. Each Drawing acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_-effectivity_select.

4.3.1260 Work_request to External_document

Each Work_request refers to zero or more External_document objects in the role of scope. Each External_document acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_-effectivity_select.

4.3.1261 Work_request to Geometric_model

Each Work_request refers to zero or more Geometric_model objects in the role of scope. Each Geometric_model acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and model_select.

4.3.1262 Work_request to Item

Each Work_request refers to zero or more Item objects in the role of scope. Each Item acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_-effectivity_select.

4.3.1263 Work_request to Item_definition_instance_relationship

Each Work_request refers to zero or more Item_definition_instance_relationship objects in the role of scope. Each Item_definition_instance_relationship acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and item_relationship_select.

4.3.1264 Work_request to Item_definition_relationship

Each Work_request refers to zero or more Item_definition_relationship objects in the role of scope. Each Item_definition_relationship acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and item_relationship_select.

4.3.1265 Work_request to Item_instance

Each Work_request refers to zero or more Item_instance objects in the role of scope. Each Item_instance acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_effectivity_select.

4.3.1266 Work_request to Item_instance_relationship

Each Work_request refers to zero or more Item_instance_relationship objects in the role of scope. Each Item_instance_relationship acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and item_relationship_select.

4.3.1267 Work_request to Item_solution

Each Work_request refers to zero or more Item_solution objects in the role of scope. Each Item_solution acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_select, and specification_information_select.

4.3.1268 Work_request to Item_version

Each Work_request refers to zero or more Item_version objects in the role of scope. Each Item_version acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_effectivity_select.

4.3.1269 Work_request to Material

Each Work_request refers to zero or more Material objects in the role of scope. Each Material acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_effectivity_select.

4.3.1270 Work_request to Multi_language_string

Each Work_request refers to zero or one Multi_language_string in the role of description. Each Multi_language_string acts as description for zero or more Work_request objects.

NOTE – This assertion is established through string_select.

4.3.1271 Work_request to Physical_instance

Each Work_request refers to zero or more Physical_instance objects in the role of scope. Each Physical_instance acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select.

4.3.1272 Work_request to Physical_instance_usage

Each Work_request refers to zero or more Physical_instance_usage objects in the role of scope. Each Physical_instance_usage acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select.

4.3.1273 Work_request to Process_plan

Each Work_request refers to zero or more Process_plan objects in the role of scope. Each Process_plan acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_-effectivity_select.

4.3.1274 Work_request to Product_class

Each Work_request refers to zero or more Product_class objects in the role of scope. Each Product_class acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.1275 Work_request to Product_component

Each Work_request refers to zero or more Product_component objects in the role of scope. Each Product_component acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.1276 Work_request to Product_function

Each Work_request refers to zero or more Product_function objects in the role of scope. Each Product_function acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.1277 Work_request to Product_specification

Each Work_request refers to zero or more Product_specification objects in the role of scope. Each Product_specification acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_-effectivity_select.

4.3.1278 Work_request to Property

Each Work_request refers to zero or more Property objects in the role of scope. Each Property acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_-effectivity_select.

4.3.1279 Work_request to Simulation_information

Each Work_request refers to zero or more Simulation_information objects in the role of scope. Each Simulation_information acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and model_select.

4.3.1280 Work_request to Specification

Each Work_request refers to zero or more Specification objects in the role of scope. Each Specification acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.1281 Work_request to Specification_category

Each Work_request refers to zero or more Specification_category objects in the role of scope. Each Specification_category acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and specification_information_select.

4.3.1282 Work_request to Specification_expression

Each Work_request refers to zero or more Specification_expression objects in the role of scope. Each Specification_expression acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and configuration_information_select.

4.3.1283 Work_request to Specification_inclusion

Each Work_request refers to zero or more Specification_inclusion objects in the role of scope. Each Specification_inclusion acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and configuration_information_select.

4.3.1284 Work_request to Styled_model

Each Work_request refers to zero or more Styled_model objects in the role of scope. Each Styled_model acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select, activity_element_and_effectivity_-select, and model_select.

4.3.1285 Work_request to View_area

Each Work_request refers to zero or more View_area objects in the role of scope. Each View_area acts as scope for zero or more Work_request objects.

NOTE – This assertion is established through activity_element_select and activity_element_and_-effectivity_select.

5 Application interpreted model

5.1 Mapping table

This clause contains the mapping table that shows how each UoF and application object of this part of ISO 10303 (see clause 4) maps to one or more AIM constructs (see annex A). The mapping table is organized in five columns.

– Column 1) Application element: Name of an application element as it appears in the application object definition in 4.2. Application object names are written in uppercase. Attribute names and assertions are listed after the application object to which they belong and are written in lower case.

– Column 2) AIM element: Name of an AIM element as it appears in the AIM (see annex A), the term 'IDENTICAL MAPPING', or the term 'PATH'. AIM entities are written in lower case. Attribute names of AIM entities are referred to as <entity name> . <attribute name>. The mapping of an application element may result in several related AIM elements. Each of these AIM elements requires a line of its own in the table. The term 'IDENTICAL MAPPING' indicates that both application objects of an application assertion map to the same AIM element. The term 'PATH' indicates that the application assertion maps to the entire reference path.

– Column 3) Source: For those AIM elements that are interpreted from the integrated resources, this is the number of the corresponding part of ISO 10303. For those AIM elements that are created for the purpose of this part of ISO 10303, this is the number of this part. For those AIM elements that are directly incorporated from an application interpreted construct (AIC) this is the AIC reference.

– Column 4) Rules: One or more numbers may be given that refer to rules that apply to the current AIM element or reference path. For rules that are derived from relationships between application objects, the same rule is referred to by the mapping entries of all the involved AIM elements. The expanded names of the rules are listed after the table.

– Column 5) Reference path: To describe fully the mapping of an ARM element, it may be necessary to specify a reference path through several related AIM elements. The reference path column documents the role of an AIM element relative to the AIM element in the row succeeding it. Two or more such related AIM elements define the interpretation of the integrated resources that satisfies the requirement specified by the application object. For each AIM element that has been created for use within this part of ISO 10303, a reference path up to its supertype from an integrated resource is specified.

For the expression of reference paths the following notational conventions apply:

a) `[]` : multiple AIM elements or sections of the reference path are required to satisfy an information requirement;

b) `()` : multiple AIM elements or sections of the reference path are identified as alternatives within the mapping to satisfy an information requirement;

c) `<>` : at least one of the enclosed sections of the reference path are required to satisfy an information requirement;

d) `{}` : enclosed section constrains the reference path to satisfy an information requirement;

e) `||` : the mapping is to the enclosed supertype entity only;

f) `->` : attribute references the entity or select type given in the following row;

g) `<-` : entity or select type is referenced by the attribute in the following row;

h) `[i]` : attribute is an aggregation of which a single member is given in the following row;

i) `[n]` : attribute is an aggregation of which member `n` is given in the following row;

j) `=>` : entity is a supertype of the entity given in the following row;

k) `<=` : entity is a subtype of the entity given in the following row;

l) `=` : the string, select, or enumeration type is constrained to a choice or value;

m) `\` : the line continuations for strings that wrap.

The entries '41', '43', '44', '519', and '520' in the third column of the mapping tables (source) refer to the following documents:

ISO 10303–41:—[3], *Industrial automation systems and integration*
– Product data representation and exchange:
– Integrated generic resource: Fundamentals of product description and support
– ISO TC 184/SC4 N539

ISO 10303–43:—[3], *Industrial automation systems and integration*
– Product data representation and exchange:
– Integrated generic resource: Representation structures
– ISO TC 184/SC4 N540

ISO 10303–44:—[3], *Industrial automation systems and integration*
– Product data representation and exchange:
– Integrated generic resource: Product structure configuration
– ISO TC 184/SC4 N541

ISO 10303–5x1:—[3], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Geometric tolerances
– ISO TC 184/SC4 N542

[3] To be published.

ISO 10303–5x2:—[3], *Industrial automation systems and integration*
– Product data representation and exchange:
– Application interpreted construct: Associative draughting elements
– ISO TC 184/SC4 N543

Table 2 – Mapping table surface_condition UoF (C1)

Application element	AIM element	Source	Rules	Reference path
COATING_LAYER	(representation) (shape_aspect)	43 41		({[representation representation.description = 'coating layer'] [representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition ->>represented_definition represented_definition = property_definition property_definition {property_definition => surface_condition} property_definition.definition ->>characterized_definition characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect shape_aspect.description = 'coating layer shape aspect']}) ({shape_aspect shape_aspect.description = 'coating layer shape aspect'})
coating_layer to value_with_unit (as thickness)	PATH			representation {representation.description = 'coating layer'} representation.items[1] ->>>representation_item => {representation_item.name = 'thickness'} measure_representation_item <= measure_with_unit {measure_with_unit => length_measure_with_unit}
coating_layer to material (as used_ material)	PATH			(representation <- {representation.description = 'coating layer'} property_definition_representation.used_representation property_definition_representation property_definition_representation.definition ->>represented_definition represented_definition = property_definition property_definition

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition => surface_condition} property_definition.definition -> characterized_definition <- {characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect shape_aspect.description = 'coating layer shape aspect'}) (shape_aspect {shape_aspect.description = 'coating layer shape aspect'} shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <-) material_designation.definitions[i] material_designation
COATING_LAYER_ RELATIONSHIP	shape_aspect_ relationship	41		{shape_aspect_relationship shape_aspect_relationship.name = 'coating layer sequence'}
description	shape_aspect_ relationship. description	41		
id	identification_ assignment.assigned_ id	41	33	shape_aspect_relationship identification_item = shape_aspect_relationship {shape_aspect_relationship.name = 'coating layer sequence'} identification_item <- applied_identification_assignment.items[1] applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'id'} identification_assignment.assigned_id
purpose	shape_aspect_ relationship. description	41		

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
coating_layer_ relationship to multi_language_ string (as description)	PATH		160	shape_aspect_relationship {shape_aspect_relationship.name = 'coating layer sequence'} multi_language_attribute_item = shape_aspect_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
coating_layer_ relationship to coating_layer (as following_ layer)	PATH			shape_aspect_relationship {shape_aspect_relationship.name = 'coating layer sequence'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'coating layer shape aspect'} shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition {property_definition => surface_condition} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {representation.description = 'coating layer'}
coating_layer_ relationship to coating_layer (as preceding_ layer)	PATH			shape_aspect_relationship {shape_aspect_relationship.name = 'coating layer sequence'} shape_aspect_relationship.relating_shape_aspect -> shape_aspect {shape_aspect.description = 'coating layer shape aspect'} shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <-

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition property_definition {property_definition => surface_condition} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {representation.description = 'coating layer'}
coating_layer_ relationship to multi_language_ string (as purpose)	PATH		160	shape_aspect_relationship {shape_aspect_relationship.name = 'coating layer sequence'} multi_language_attribute_item = shape_aspect_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
CONTACT_RATIO	representation	43		{[representation representation.description = 'contact ratio'] [representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition => surface_condition]}
ratio_value	measure_ representation_item	45		representation {representation.description = 'contact ratio'} representation.items[i] -> representation_item => {representation_item.name = 'ratio value'}

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
				measure_representation_item {measure_representation_item <= measure_with_unit => ratio_measure_with_unit}
contact_ratio to physical_model (as measuring_reference)	PATH			representation <- {representation.description = 'contact ratio'} property_definition_representation.used_representation property_definition_representation {property_definition_representation.name = 'measuring reference'} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition => surface_condition} property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition => product_definition_with_associated_documents => physically_modelled_product_definition
DEFAULT_ ROUGHNESS	representation	43		{representation [representation.description = 'default roughness'] [representation.context_of_items -> representation_context representation_context.context_type = 'default setting']}
description	descriptive_ representation_item	45		representation {representation.description = 'default roughness'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
default_ roughness to multi_language_ string (as description)	PATH		101	representation {representation.description = 'default roughness'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
default_ roughness to value_with_unit (as max_value)	PATH			representation {representation.description = 'default roughness'} representation.items[i] -> representation_item => {representation_item.name = 'max value'} measure_representation_item <= measure_with_unit
default_ roughness to value_with_unit (as min_value)	PATH			representation {representation.description = 'default roughness'} representation.items[i] -> representation_item => {representation_item.name = 'min value'} measure_representation_item <= measure_with_unit
DEFAULT_SETTING_ ASSOCIATION	(property_definition_ representation) (representation_ relationship)	41 43		({property_definition_representation property_definition_representation.name = 'default setting'}) ({representation_relationship representation_relationship.name = 'default setting'})

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
default_setting_ association to digital_document (as default_ context)	PATH		27	representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_2 ->
representation =>
externally_defined_representation digital_document_reference_item = externally_defined_representation digital_document_reference_item <- digital_document_reference.items[1] digital_document_reference <= document_reference {document_reference.role -> document_role document_role.name = 'external definition'} document_reference.assigned_document -> document {document.kind -> document_type document_type.product_data_type = 'digital'}
default_setting_ association to drawing_sheet (as default_ context)	PATH			representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_2 ->
representation =>
presentation_representation =>
presentation_area =>
drawing_sheet_revision
default_setting_ association to item_shape (as default_ context)	PATH			property_definition_representation {property_definition_representation.name = 'default setting'} property_definition_representation.definition ->
represented_definition = property_definition property_definition =>
product_definition_shape

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
default_setting_ association to view_area (as default_ context)	PATH			representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_2 ->>>representation => presentation_representation => (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)
default_setting_ association to default_ roughness (as default_ value)	PATH			(property_definition_representation {property_definition_representation.name = 'default setting'} property_definition_representation.used_representation ->) (representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_1 ->) representation {representation.description = 'default roughness'}
HARDNESS	representation	43		{[representation representation.description = 'hardness'] [representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition ->>>represented_definition represented_definition = property_definition property_definition => surface_condition]}
description	descriptive_ representation_item	45		representation {representation.description = 'hardness'} representation.items[i] ->>>representation_item => {representation_item.name = 'description'} descriptive_representation_item
environment_ condition	data_environment	45		representation <- {representation.description = 'hardness'} property_definition_representation.used_representation

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation => material_property_representation material_property_representation.dependent_environment -> data_environment
measuring_method	descriptive_ representation_item	45		representation {representation.description = 'hardness'} representation.items[i] -> representation_item => {representation_item.name = 'measuring method'} descriptive_representation_item
measuring_ position	descriptive_ representation_item	45		representation {representation.description = 'hardness'} representation.items[i] -> representation_item => {representation_item.name = 'measuring position'} descriptive_representation_item
hardness to value_with_unit (as depth)	PATH			representation {representation.description = 'hardness'} representation.items[i] -> representation_item => {representation_item.name = 'depth'} measure_representation_item <= measure_with_unit {measure_with_unit => length_measure_with_unit}
hardness to multi_language_ string (as description)	PATH		101	representation {representation.description = 'hardness'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <=

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
				multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
hardness to value_with_unit (as hardness_ value)	PATH			representation {representation.description = 'hardness'} representation.items[i] -> representation_item => {representation_item.name = 'hardness value'} measure_representation_item <= measure_with_unit
ROUGHNESS	representation	43		{[representation representation.description = 'roughness'] [representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition => surface_condition]}
measuring_method	descriptive_ representation_item	45		representation {representation.description = 'roughness'} representation.items[i] -> representation_item => {representation_item.name = 'measuring method'} descriptive_representation_item
roughness to value_with_unit (as cutoff_ value)	PATH			representation {representation.description = 'roughness'} representation.items[i] -> representation_item => {representation_item.name = 'cutoff value'} measure_representation_item <= measure_with_unit {measure_with_unit => length_measure_with_unit}

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
roughness to curve_on_surface (as measuring_direction)	PATH			representation {representation.description = 'roughness'} representation.items[i] -> representation_item => {representation_item.name = 'measuring direction'} geometric_representation_item => curve => (pcurve) (surface_curve)
roughness to value_with_unit (as roughness_max_value)	PATH			representation {representation.description = 'roughness'} representation.items[i] -> representation_item => {representation_item.name = 'roughness max value'} measure_representation_item <= measure_with_unit
roughness to value_with_unit (as roughness_min_value)	PATH			representation {representation.description = 'roughness'} representation.items[i] -> representation_item => {representation_item.name = 'roughness min value'} measure_representation_item <= measure_with_unit
roughness to value_with_unit (as sampling_length)	PATH			representation {representation.description = 'roughness'} representation.items[i] -> representation_item => {representation_item.name = 'sampling length'} measure_representation_item <= measure_with_unit {measure_with_unit => length_measure_with_unit}
SURFACE_CONDITION	representation	43		{[representation (representation.description = 'coating layer') (representation.description = 'contact ratio') (representation.description = 'hardness')

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
				(representation.description = 'roughness') (representation.description = 'tactile appearance') (representation.description = 'treatment result') (representation.description = 'visual appearance')] [representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition => surface_condition]}
value_ interpretation	representation_ context.context_type	43		representation representation.context_of_items -> representation_context representation_context.context_type {(representation_context.context_type) (representation_context.context_type = 'required') (representation_context.context_type= 'calculated') (representation_context.context_type= 'measured') (representation_context.context_type= 'estimated')}
surface_ condition to item_shape (as is_applied_ to)	PATH			(representation <- {(representation.description = 'contact ratio') (representation.description = 'hardness') (representation.description = 'roughness') (representation.description = 'tactile appearance') (representation.description = 'treatment result') (representation.description = 'visual appearance')} property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition =>

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
				surface_condition} property_definition.definition ->
characterized_definition characterized_definition = shape_definition shape_definition = product_definition_shape) (representation <- {representation.description = 'coating layer'} property_definition_representation.used_representation property_definition_representation property_definition_representation.definition ->
represented_definition represented_definition = property_definition property_definition {property_definition => surface_condition} property_definition.definition ->
characterized_definition characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect {shape_aspect.description = 'coating layer shape aspect'} shape_aspect.of_shape ->) product_definition_shape
surface_condition to shape_aspect (as is_applied_to)	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition ->
represented_definition represented_definition = property_definition property_definition {property_definition => surface_condition} property_definition.definition ->
characterized_definition

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect
surface_ condition to process_ operation (as is_caused_ by)	PATH			representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition => surface_condition} property_or_shape_select = property_definition property_or_shape_select <- process_property_association.property_or_shape process_property_association process_property_association.process -> property_process <= action {action.description = 'process operation'}
TACTILE_ APPEARANCE	representation	43		{[representation representation.description = 'tactile appearance'] [representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition => surface_condition]}
description	descriptive_ representation_item	45		representation {representation.description = 'tactile appearance'} representation.items[i] -> representation_item => {representation_item.name = 'description'}

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
				descriptive_representation_item
id	representation.id	43		
name	representation.name	43		
tactile_ appearance to value_with_unit (as depth)	measure_ representation_item	45		representation {representation.description = 'tactile appearance'} representation.items[i] -> representation_item => {representation_item.name = 'depth'} measure_representation_item <= measure_with_unit {measure_with_unit => length_measure_with_unit}
tactile_ appearance to multi_language_ string (as description)	PATH		101	representation {representation.description = 'tactile appearance'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
tactile_ appearance to multi_language_ string (as name)	PATH		153	representation {representation.description = 'tactile appearance'} multi_language_attribute_item = representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
TREATMENT_RESULT	representation	43		{[representation representation.description = 'treatment result']

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
				[representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition => surface_condition]}
description	descriptive_ representation_item	45		representation {representation.description = 'treatment result'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item
id	representation.id	43		
name	representation.name	43		
purpose	descriptive_ representation_item	45		representation {representation.description = 'treatment result'} representation.items[i] -> representation_item => {representation_item.name = 'purpose'} descriptive_representation_item
treatment_result to multi_ language_string (as description)	PATH		101	representation {representation.description = 'treatment result'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
				multi_language_attribute_assignment.attribute_name = 'description'}
treatment_result to multi_ language_string (as name)	PATH		153	representation {representation.description = 'treatment result'} multi_language_attribute_item = representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
treatment_result to multi_ language_string (as purpose)	PATH		101	representation {representation.description = 'treatment result'} representation.items[i] -> representation_item => {representation_item.name = 'purpose'} descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
VALUE_WITH_UNIT	measure_with_unit	41	35	
unit_component	(named_unit) (derived_unit)	41 41	25	measure_with_unit measure_with_unit.unit_component -> (unit = named_unit named_unit) (unit = derived_unit derived_unit)
value_component	measure_value	41		measure_with_unit measure_with_unit.value_component -> measure_value
VISUAL_ APPEARANCE	representation	43		{[representation representation.description = 'visual appearance']

Table 2 – Mapping table surface_condition UoF (C1) (continued)

Application element	AIM element	Source	Rules	Reference path
				[representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition => surface_condition]}
colour_id	descriptive_ representation_item	45		representation {representation.description = 'visual appearance'} representation.items[i] -> representation_item => {representation_item.name = 'colour id'} descriptive_representation_item
colour_name	descriptive_ representation_item	45		representation {representation.description = 'visual appearance'} representation.items[i] -> representation_item => {representation_item.name = 'colour name'} descriptive_representation_item
id	representation.id	43		
lustre	descriptive_ representation_item	45		representation {representation.description = 'visual appearance'} representation.items[i] -> representation_item => {representation_item.name = 'lustre'} descriptive_representation_item
name	representation.name	43		
pattern	descriptive_ representation_item	45		representation {representation.description = 'visual appearance'} representation.items[i] -> representation_item => {representation_item.name = 'pattern'} descriptive_representation_item

Table 2 – Mapping table surface_condition UoF (C1) (concluded)

Application element	AIM element	Source	Rules	Reference path
transparency	descriptive_ representation_item	45		representation {representation.description = 'visual appearance'} representation.items[i] -> representation_item => {representation_item.name = 'transparency'} descriptive_representation_item
visual_ appearance to multi_language_ string (as colour_name)	PATH		101	representation {representation.description = 'visual appearance'} representation.items[i] -> representation_item => {representation_item.name = 'colour name'} descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
visual_ appearance to multi_language_ string (as name)	PATH		153	representation {representation.description = 'visual appearance'} multi_language_attribute_item = representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}

Table 3 – Mapping table explicit_draughting UoF (D1)

Application element	AIM element	Source	Rules	Reference path
ANNOTATION_CURVE	annotation_curve_ occurrence	504		
annotation_curve_ type	curve_style.name	46		annotation_curve_occurrence <= annotation_occurrence <= {annotation_occurrence => draughting_annotation_occurrence} styled_item styled_item.styles[i] -> presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select presentation_style_select = curve_style curve_style curve_style.name
DETAILED_ELEMENT	representation_item	43		{representation_item => (styled_item) (mapped_item)}
name	representation_item. name	43		
detailed_element to multi_ language_string (as name)	PATH		154	representation_item multi_language_attribute_item = representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
DIMENSION	(angular_dimension) (curve_dimension) (diameter_dimension) (leader_directed_ dimension) (linear_dimension) (ordinate_dimension)	506 506 506 506 506 506		(angular_dimension <= dimension_curve_directed_callout) (curve_dimension <= dimension_curve_directed_callout) (diameter_dimension <= dimension_curve_directed_callout) (leader_directed_dimension <= leader_directed_callout) (linear_dimension <=

Table 3 – Mapping table explicit_draughting UoF (D1) (continued)

Application element	AIM element	Source	Rules	Reference path
	(radius_dimension)	506		dimension_curve_directed_callout) (ordinate_dimension <= projection_directed_callout) (radius_dimension <= dimension_curve_directed_callout)
id	representation_item. name	43		(angular_dimension <= dimension_curve_directed_callout <=) (curve_dimension <= dimension_curve_directed_callout <=) (diameter_dimension <= dimension_curve_directed_callout <=) (leader_directed_dimension <= leader_directed_callout <=) (linear_dimension <= dimension_curve_directed_callout <=) (ordinate_dimension <= projection_directed_callout <=) (radius_dimension <= dimension_curve_directed_callout <=) draughting_callout <= {draughting_callout => draughting_elements} geometric_representation_item <= representation_item representation_item.name
dimension to annotation_curve (as composition_ curve)	PATH			(angular_dimension <= dimension_curve_directed_callout <=) (curve_dimension <= dimension_curve_directed_callout <=) (diameter_dimension <= dimension_curve_directed_callout <=) (leader_directed_dimension <= leader_directed_callout <=) (linear_dimension <= dimension_curve_directed_callout <=) (ordinate_dimension <=

Table 3 – Mapping table explicit_draughting UoF (D1) (continued)

Application element	AIM element	Source	Rules	Reference path
				projection_directed_callout <=) (radius_dimension <= dimension_curve_directed_callout <=) draughting_callout draughting_callout.contents[i] -> draughting_callout_element draughting_callout_element = annotation_curve_occurrence annotation_curve_occurrence
dimension to dimension_ callout (as primary_ callout)	PATH			(angular_dimension <= dimension_curve_directed_callout <=) (curve_dimension <= dimension_curve_directed_callout <=) (diameter_dimension <= dimension_curve_directed_callout <=) (leader_directed_dimension <= leader_directed_callout <=) (linear_dimension <= dimension_curve_directed_callout <=) (ordinate_dimension <= projection_directed_callout <=) (radius_dimension <= dimension_curve_directed_callout <=) draughting_callout <- {draughting_callout => draughting_elements} draughting_callout_relationship.relating_draughting_callout draughting_callout_relationship {draughting_callout_relationship => dimension_callout_relationship} {draughting_callout_relationship.name = 'primary'} draughting_callout_relationship.related_draughting_callout -> (draughting_callout) (draughting_callout => structured_dimension_callout)

Table 3 – Mapping table explicit_draughting UoF (D1) (continued)

Application element	AIM element	Source	Rules	Reference path
dimension to dimension_ callout (as secondary_ callout)	PATH			(angular_dimension <= dimension_curve_directed_callout <=) (curve_dimension <= dimension_curve_directed_callout <=) (diameter_dimension <= dimension_curve_directed_callout <=) (leader_directed_dimension <= leader_directed_callout <=) (linear_dimension <= dimension_curve_directed_callout <=) (ordinate_dimension <= projection_directed_callout <=) (radius_dimension <= dimension_curve_directed_callout <=) draughting_callout <- {draughting_callout => draughting_elements} draughting_callout_relationship.relating_draughting_callout draughting_callout_relationship {draughting_callout_relationship => dimension_callout_relationship} {draughting_callout_relationship.name = 'secondary'} draughting_callout_relationship.related_draughting_callout -> (draughting_callout) (draughting_callout => structured_dimension_callout)
DIMENSION_ CALLOUT	(draughting_callout) (structured_ dimension_callout)	506 506		structured_dimension_callout <= draughting_callout
DIMENSION_CURVE	dimension_curve	506		
dimension_curve to annotation_ symbol (as terminator)	PATH		41	dimension_curve <= annotation_curve_occurrence <- terminator_symbol.annotated_curve terminator_symbol <= {terminator_symbol =>

Table 3 – Mapping table explicit_draughting UoF (D1) (continued)

Application element	AIM element	Source	Rules	Reference path
				dimension_curve_terminator} annotation_symbol_occurrence {annotation_symbol_occurrence <= annotation_occurrence <= {annotation_occurrence => draughting_annotation_occurrence} styled_item styled_item.item -> representation_item => geometric_representation_item => defined_symbol defined_symbol.definition -> defined_symbol_select defined_symbol_select = pre_defined_symbol pre_defined_symbol => pre_defined_terminator_symbol}
DIMENSION_ SEQUENCE_PAIR	dimension_pair	506		dimension_pair <= draughting_callout_relationship {(draughting_callout_relationship.name = 'chained') (draughting_callout_relationship.name = 'parallel')}
dimension_ sequence_pair to dimension (as predecessor)	PATH			dimension_pair <= draughting_callout_relationship draughting_callout_relationship.relating_draughting_callout -> draughting_callout => {draughting_callout => draughting_elements} (dimension_curve_directed_callout => angular_dimension) (dimension_curve_directed_callout => curve_dimension) (dimension_curve_directed_callout => diameter_dimension) (dimension_curve_directed_callout => linear_dimension) (projection_directed_callout => ordinate_dimension)

Table 3 – Mapping table explicit_draughting UoF (D1) (continued)

Application element	AIM element	Source	Rules	Reference path
				(dimension_curve_directed_callout => radius_dimension)
dimension_ sequence_pair to dimension (as successor)	PATH			dimension_pair <= draughting_callout_relationship draughting_callout_relationship.related_draughting_callout ->
draughting_callout => {draughting_callout => draughting_elements} (dimension_curve_directed_callout => angular_dimension) (dimension_curve_directed_callout => curve_dimension) (dimension_curve_directed_callout => diameter_dimension) (dimension_curve_directed_callout => linear_dimension) (projection_directed_callout => ordinate_dimension) (dimension_curve_directed_callout => radius_dimension)
DRAWING	auto_design_drawing_ revision	214	192	{auto_design_drawing_revision <= drawing_revision}
drawing_id	drawing_definition. drawing_number	101		auto_design_drawing_revision <= drawing_revision drawing_revision.drawing_identifier ->
drawing_definition drawing_definition.drawing_number
drawing_revision_ id	drawing_revision. revision_identifier	101		auto_design_drawing_revision <= drawing_revision drawing_revision.revision_identifier
drawing_ specification	document.id	41		auto_design_drawing_revision <= drawing_revision specified_item = drawing_revision specified_item <- draughting_specification_reference.specified_items[i] draughting_specification_reference <=

Table 3 – Mapping table explicit_draughting UoF (D1) (continued)

Application element	AIM element	Source	Rules	Reference path
				document_reference document_reference.assigned_document ->
document {document.kind ->
document_type document_type.product_data_type = 'draughting specification'} document.id
language	draughting_title. language	101		auto_design_drawing_revision <= drawing_revision draughting_titled_item = drawing_revision draughting_titled_item <- draughting_title.items[i] draughting_title draughting_title.language
title	draughting_title. contents	101		auto_design_drawing_revision <= drawing_revision draughting_titled_item = drawing_revision draughting_titled_item <- draughting_title.items[i] draughting_title draughting_title.contents
drawing to document_type (as drawing_ type)	PATH			auto_design_drawing_revision <= drawing_revision drawing_revision.drawing_identifier ->
drawing_definition drawing_definition.drawing_type
drawing to multi_ language_string (as title)	PATH		107	auto_design_drawing_revision <= drawing_revision draughting_titled_item = drawing_revision draughting_titled_item <- draughting_title.items[i] draughting_title multi_language_attribute_item = draughting_title multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment

Table 3 – Mapping table explicit_draughting UoF (D1) (continued)

Application element	AIM element	Source	Rules	Reference path
				{applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'contents'}
DRAWING_ ASSIGNMENT	applied_presented_ item	214		applied_presented_item <= presented_item
drawing_ assignment to drawing (as drawing)	PATH			applied_presented_item <= presented_item <- presented_item_representation.item presented_item_representation presented_item_representation.presentation -> presentation_representation_select presentation_representation_select = presentation_set presentation_set => drawing_revision => auto_design_drawing_revision
drawing_ assignment to class_function_ association (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select = configuration_design configuration_design {configuration_design configuration_design.configuration -> configuration_item configuration_item.description = 'class function definition'}
drawing_ assignment to design_ constraint (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select = representation representation {representation.description = 'design constraint'}

Table 3 – Mapping table explicit_draughting UoF (D1) (continued)

Application element	AIM element	Source	Rules	Reference path
drawing_ assignment to design_ discipline_item_ definition (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] ->\ presented_item_select = product_definition product_definition
drawing_ assignment to item_definition_ instance_ relationship (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] ->\ presented_item_select = product_definition_relationship product_definition_relationship
drawing_ assignment to item_definition_ relationship (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] ->\ presented_item_select = product_definition_relationship product_definition_relationship
drawing_ assignment to item_instance (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] ->\ (presented_item_select = product_definition_occurrence product_definition_occurrence) (presented_item_select = assembly_component_usage assembly_component_usage)
drawing_ assignment to item_instance_ relationship (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] ->\ presented_item_select (presented_item_select = product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition =>\ product_definition_occurrence}) (presented_item_select = product_definition_substitute

Table 3 – Mapping table explicit_draughting UoF (D1) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_substitute)
drawing_ assignment to item_solution (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] ->
presented_item_select = solution_based_product_definition solution_based_product_definition
drawing_ assignment to process_ operation (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] ->
presented_item_select = action action {action action.description = 'process operation'}
drawing_ assignment to process_plan (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] ->
presented_item_select = process_plan process_plan
drawing_ assignment to product_class (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] ->
presented_item_select = product_class product_class
drawing_ assignment to product_ component (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] ->
presented_item_select = product_component product_component
drawing_ assignment to product_function (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] ->
presented_item_select = product_definition product_definition

Table 3 – Mapping table explicit_draughting UoF (D1) (continued)

Application element	AIM element	Source	Rules	Reference path
drawing_ assignment to specification (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> (presented_item_select = product_concept_feature product_concept_feature) (presented_item_select = package_product_concept_feature package_product_concept_feature)
drawing_ assignment to specification_ category (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> (presented_item_select = product_concept_feature_category product_concept_feature_category) (presented_item_select = exclusive_product_concept_feature_category exclusive_product_concept_feature_category)
DRAWING_SEQUENCE	drawing_revision_ sequence	101		
drawing_sequence to drawing (as predecessor)	PATH			drawing_revision_sequence drawing_revision_sequence.predecessor -> drawing_revision => auto_design_drawing_revision
drawing_sequence to drawing (as successor)	PATH			drawing_revision_sequence drawing_revision_sequence.successor -> drawing_revision => auto_design_drawing_revision
DRAWING_SHEET	drawing_sheet_ revision	101		
sheet_id	drawing_sheet_ revision_usage.sheet_ number	101		drawing_sheet_revision <= presentation_area <- area_in_set.area area_in_set => drawing_sheet_revision_usage drawing_sheet_revision_usage.sheet_number
sheet_revision_ id	drawing_sheet_ revision.revision_ identifier	101		
size	presentation_size. size	46		drawing_sheet_revision <= presentation_area presentation_size_assignment_select = presentation_area

Table 3 – Mapping table explicit_draughting UoF (D1) (continued)

Application element	AIM element	Source	Rules	Reference path
				presentation_size_assignment_select <- presentation_size.unit presentation_size presentation_size.size
title	representation.name	43		drawing_sheet_revision <= presentation_area <= presentation_representation <= representation representation.name
drawing_sheet to drawing (as associated_ drawing)	PATH		188	drawing_sheet_revision <= presentation_area <- area_in_set.area area_in_set {area_in_set => drawing_sheet_revision_usage} area_in_set.in_set -> presentation_set => drawing_revision => auto_design_drawing_revision
drawing_sheet to cartesian_ coordinate_space_ 2d (as sheet_space)	PATH			drawing_sheet_revision <= presentation_area <= presentation_representation <= representation representation.context_of_items -> representation_context => geometric_representation_context {geometric_representation_context.coordinate_space_dimension = 2}
drawing_sheet to multi_language_ string (as title)	PATH		108	drawing_sheet_revision multi_language_attribute_item = drawing_sheet_revision multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'representation.name'}

Table 3 – Mapping table explicit_draughting UoF (D1) (continued)

Application element	AIM element	Source	Rules	Reference path
PROJECTION_CURVE	projection_curve	506		
VIEW_AREA_ ASSIGNMENT	applied_presented_ item	214		
view_area_ assignment to class_function_ association (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select = configuration_design
view_area_ assignment to design_ constraint (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select = representation representation {representation.description = 'design constraint'}
view_area_ assignment to design_ discipline_item_ definition (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select = product_definition
view_area_ assignment to item_definition_ instance_ relationship (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select = product_definition_relationship
view_area_ assignment to item_definition_ relationship (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select = product_definition_relationship

Table 3 – Mapping table explicit_draughting UoF (D1) (continued)

Application element	AIM element	Source	Rules	Reference path
view_area_ assignment to item_instance (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> (presented_item_select = product_definition_occurrence) (presented_item_select = assembly_component_usage)
view_area_ assignment to item_instance_ relationship (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select (presented_item_select = product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}) (presented_item_select = product_definition_substitute product_definition_substitute)
view_area_ assignment to item_solution (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select = solution_based_product_definition
view_area_ assignment to process_ operation (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select = action action {action.description = 'process operation'}
view_area_ assignment to process_plan (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select = process_plan

Table 3 – Mapping table explicit_draughting UoF (D1) (concluded)

Application element	AIM element	Source	Rules	Reference path
view_area_ assignment to product_class (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select = product_class
view_area_ assignment to product_ component (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select = product_component
view_area_ assignment to product_function (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> presented_item_select = product_definition
view_area_ assignment to specification (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> (presented_item_select = product_concept_feature) (presented_item_select = package_product_concept_feature)
view_area_ assignment to specification_ category (as is_ describing)	PATH			applied_presented_item applied_presented_item.items[i] -> (presented_item_select = product_concept_feature_category) (presented_item_select = exclusive_product_concept_feature_category)
view_area_ assignment to view_area (as view_area)	PATH			applied_presented_item <= presented_item <- presented_item_representation.item presented_item_representation presented_item_representation.presentation -> presentation_representation_select presentation_representation_select = presentation_set presentation_set

Table 4 – Mapping table associative_annotation UoF (D2)

Application element	AIM element	Source	Rules	Reference path
ASSOCIATED_ ANNOTATION_ SYMBOL	draughting_elements	506		draughting_elements <= draughting_callout
associated_ annotation_ symbol to annotation_curve (as is_ associated_to)	PATH			draughting_elements <= draughting_callout draughting_callout.contents[i] -> draughting_callout_element draughting_callout_element = annotation_curve_occurrence annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence => {annotation_occurrence => draughting_annotation_occurrence} annotation_curve_occurrence
associated_ annotation_ symbol to annotation_ subfigure (as is_ associated_to)	PATH			draughting_elements <= draughting_callout draughting_callout.contents[i] -> draughting_callout_element draughting_callout_element = annotation_curve_occurrence annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence =>

Table 4 – Mapping table associative_annotation UoF (D2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{annotation_occurrence => draughting_annotation_occurrence} annotation_symbol_occurrence => annotation_subfigure_occurrence
associated_ annotation_ symbol to annotation_ symbol (as is_ associated_to)	PATH			draughting_elements <= draughting_callout draughting_callout.contents[i] -> draughting_callout_element draughting_callout_element = annotation_curve_occurrence annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence {annotation_occurrence => draughting_annotation_occurrence} (annotation_occurrence => annotation_symbol_occurrence) (annotation_occurrence <= styled_item styled_item.item -> representation_item => geometric_representation_item => draughting_callout => draughting_elements)

Table 4 – Mapping table associative_annotation UoF (D2) (continued)

Application element	AIM element	Source	Rules	Reference path
associated_ annotation_ symbol to detailed_ geometric_model_ element (as is_ associated_to)	PATH			draughting_elements <= draughting_callout draughting_callout.contents[i] -> draughting_callout_element draughting_callout_element = annotation_curve_occurrence annotation_curve_occurrence <= annotation_occurrence <= {annotation_occurrence => draughting_annotation_occurrence} styled_item styled_item.item -> representation_item <- {representation_item => geometric_representation_item => curve} representation.items[1] representation {representation => shape_representation} {representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship => shape_aspect_associativity} representation.items[2] -> representation_item

Table 4 – Mapping table associative_annotation UoF (D2) (continued)

Application element	AIM element	Source	Rules	Reference path
associated_ annotation_ symbol to fill_ area (as is_ associated_to)	PATH			draughting_elements <= draughting_callout draughting_callout.contents[i] -> draughting_callout_element draughting_callout_element = annotation_curve_occurrence annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence => {annotation_occurrence => draughting_annotation_occurrence} annotation_fill_area_occurrence
associated_ annotation_ symbol to text (as is_ associated_to)	PATH			draughting_elements <= draughting_callout draughting_callout.contents[i] -> draughting_callout_element draughting_callout_element = annotation_curve_occurrence annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence => {annotation_occurrence => draughting_annotation_occurrence} annotation_text_occurrence

Table 4 – Mapping table associative_annotation UoF (D2) (continued)

Application element	AIM element	Source	Rules	Reference path
ASSOCIATED_FILL_AREA	annotation_fill_area_occurrence	504		
associated_fill_area to annotation_curve (as is_derived_from)	PATH			annotation_fill_area_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence => {annotation_occurrence => draughting_annotation_occurrence} annotation_curve_occurrence
associated_fill_area to detailed_geometric_model_element (as is_derived_from)	PATH			annotation_fill_area_occurrence <= annotation_occurrence <= {annotation_occurrence => draughting_annotation_occurrence} styled_item styled_item.item -> representation_item => geometric_representation_item => annotation_fill_area annotation_fill_area.boundaries[i] -> curve <= geometric_representation_item <= representation_item <- representation.items[1] representation {representation => shape_representation} {representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition ->

Table 4 – Mapping table associative_annotation UoF (D2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition ->& characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship => shape_aspect_associativity} representation.items[2] ->& representation_item
ASSOCIATED_ LEADER	leader_curve	506		
associated_ leader to annotation_curve (as target)	PATH			leader_curve <= annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence ->& annotation_occurrence => {annotation_occurrence => draughting_annotation_occurrence} annotation_curve_occurrence
associated_ leader to annotation_ subfigure (as target)	PATH			leader_curve <= annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence ->&

Table 4 – Mapping table associative_annotation UoF (D2) (continued)

Application element	AIM element	Source	Rules	Reference path
				annotation_occurrence => {annotation_occurrence => draughting_annotation_occurrence} annotation_symbol_occurrence => annotation_subfigure_occurrence
associated_ leader to annotation_ symbol (as target)	PATH			leader_curve <= annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence {annotation_occurrence => draughting_annotation_occurrence} (annotation_occurrence => annotation_symbol_occurrence) (annotation_occurrence <= styled_item styled_item.item -> representation_item => geometric_representation_item => draughting_callout => draughting_elements)
associated_ leader to detailed_ geometric_model_ element (as target)	PATH			leader_curve <= annotation_curve_occurrence <= annotation_occurrence <= {annotation_occurrence => draughting_annotation_occurrence} styled_item styled_item.item -> representation_item <- {representation_item =>

Table 4 – Mapping table associative_annotation UoF (D2) (continued)

Application element	AIM element	Source	Rules	Reference path
				geometric_representation_item => curve} representation.items[1] representation {representation => shape_representation} {representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship => shape_aspect_associativity} representation.items[2] -> representation_item
associated_ leader to fill_ area (as target)	PATH			leader_curve <= annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence => {annotation_occurrence => draughting_annotation_occurrence} annotation_fill_area_occurrence

Table 4 – Mapping table associative_annotation UoF (D2) (continued)

Application element	AIM element	Source	Rules	Reference path
associated_ leader to text (as target)	PATH			leader_curve <= annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence => {annotation_occurrence => draughting_annotation_occurrence} annotation_text_occurrence
ASSOCIATED_ PROJECTION_CURVE	projection_curve	506		
associated_ projection_curve to annotation_ curve (as is_projected_ to)	PATH			projection_curve <= annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence => {annotation_occurrence => draughting_annotation_occurrence} annotation_curve_occurrence

Table 4 – Mapping table associative_annotation UoF (D2) (continued)

Application element	AIM element	Source	Rules	Reference path
associated_ projection_curve to annotation_ subfigure (as is_projected_ to)	PATH			projection_curve <= annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence => {annotation_occurrence => draughting_annotation_occurrence} annotation_symbol_occurrence => annotation_subfigure_occurrence
associated_ projection_curve to annotation_ symbol (as is_projected_ to)	PATH			projection_curve <= annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence {annotation_occurrence => draughting_annotation_occurrence} (annotation_occurrence => annotation_symbol_occurrence) (annotation_occurrence <= styled_item styled_item.item -> representation_item => geometric_representation_item => draughting_callout =>

Table 4 – Mapping table associative_annotation UoF (D2) (continued)

Application element	AIM element	Source	Rules	Reference path
				draughting_elements)
associated_ projection_curve to detailed_ geometric_model_ element (as is_projected_ to)	PATH			projection_curve <= annotation_curve_occurrence <= annotation_occurrence <= {annotation_occurrence => draughting_annotation_occurrence} styled_item styled_item.item -> representation_item <- {representation_item => geometric_representation_item => curve} representation.items[1] representation {representation => shape_representation} {representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> property_definition property_definition.definition -> characterized_definition characterized_definition = shape_definition shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship => shape_aspect_associativity} representation.items[2] -> representation_item
associated_ projection_curve to fill_area (as is_projected_ to)	PATH			projection_curve <= annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence

Table 4 – Mapping table associative_annotation UoF (D2) (continued)

Application element	AIM element	Source	Rules	Reference path
				annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence => {annotation_occurrence => draughting_annotation_occurrence} annotation_fill_area_occurrence
associated_ projection_curve to text (as is_projected_ to)	PATH			projection_curve <= annotation_curve_occurrence <= annotation_occurrence <- {annotation_occurrence => draughting_annotation_occurrence} annotation_occurrence_relationship.related_annotation_occurrence annotation_occurrence_relationship {annotation_occurrence_relationship => annotation_occurrence_associativity} annotation_occurrence_relationship.relating_annotation_occurrence -> annotation_occurrence => {annotation_occurrence => draughting_annotation_occurrence} annotation_text_occurrence
DETAILED_ELEMENT	representation_item	43		{representation_item => (styled_item) (mapped_item)}
name	representation_item. name	43		
detailed_element to multi_ language_string (as name)	PATH		154	representation_item multi_language_attribute_item = representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}

Table 4 – Mapping table associative_annotation UoF (D2) (continued)

Application element	AIM element	Source	Rules	Reference path
DIMENSION_ CALLOUT	(draughting_callout) (structured_ dimension_callout)	506 506		structured_dimension_callout <= draughting_callout
dimension_ callout to geometric_ dimension (as is_ displaying)	PATH			(structured_dimension_callout <= draughting_callout <=) (draughting_callout <=) {draughting_callout draughting_callout.contents[i] -> draughting_callout_element draughting_callout_element = annotation_text_occurrence annotation_text_occurrence <= annotation_occurrence <= styled_item styled_item.item -> representation_item => geometric_representation_item => text_literal => dimension_text_associativity} geometric_representation_item <= representation_item <- mapped_item.mapping_target mapped_item {mapped_item => dimension_text_associativity <= text_literal} mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => shape_representation => shape_dimension_representation <- dimensional_characteristic_representation.representation dimensional_characteristic_representation

Table 4 – Mapping table associative_annotation UoF (D2) (concluded)

Application element	AIM element	Source	Rules	Reference path
				dimensional_characteristic_representation.dimension -> dimensional_characteristic (dimensional_characteristic = dimensional_location dimensional_location) (dimensional_characteristic = dimensional_size dimensional_size)
GEOMETRIC_ TOLERANCE_SYMBOL	geometrical_ tolerance_callout	506		geometrical_tolerance_callout <= draughting_callout {draughting_callout => draughting_elements}
SURFACE_ CONDITION_SYMBOL	surface_condition_ callout	214		surface_condition_callout <= draughting_callout {draughting_callout => draughting_elements}

Table 5 – Mapping table external_reference_mechanism UoF (E1)

Application element	AIM element	Source	Rules	Reference path
ARCHIVED_ DOCUMENT	document	41		{document document.kind -> document_type document_type.product_data_type = 'physical'}
archiving_format	document_usage_ constraint.subject_ element_value	41		document <- {document.kind -> document_type document_type.product_data_type = 'physical'} document_usage_constraint.source document_usage_constraint {document_usage_constraint.subject_element = 'format'} document_usage_constraint.subject_element_value
found_in	[identification_ assignment.assigned_ id] [external_source. source_id]	41 41	33	document {document.kind -> document_type document_type.product_data_type = 'physical'} external_identification_item = document external_identification_item <- applied_external_identification_assignment.items[1] applied_external_identification_assignment <= [external_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'object name'} identification_assignment.assigned_id] [external_identification_assignment external_identification_assignment.source -> external_source {external_source.description = 'archive'} external_source.source_id]
DIGITAL_DOCUMENT	document	41		{document document.kind -> document_type document_type.product_data_type = 'digital'}

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
application_ domain	document_usage_ constraint.subject_ element_value	41		document <- {document.kind -> document_type document_type.product_data_type = 'digital'} document_usage_constraint.source document_usage_constraint {document_usage_constraint.subject_element = 'application domain'} document_usage_constraint.subject_element_value
character_code	document_usage_ constraint.subject_ element_value	41		document <- {document.kind -> document_type document_type.product_data_type = 'digital'} document_usage_constraint.source document_usage_constraint {document_usage_constraint.subject_element = 'character code'} document_usage_constraint.subject_element_value
creating_ interface	document_usage_ constraint.subject_ element_value	41		document <- {document.kind -> document_type document_type.product_data_type = 'digital'} document_usage_constraint.source document_usage_constraint {document_usage_constraint.subject_element = 'creating interface'} document_usage_constraint.subject_element_value
creating_system	document_usage_ constraint.subject_ element_value	41		document <- {document.kind -> document_type document_type.product_data_type = 'digital'} document_usage_constraint.source document_usage_constraint {document_usage_constraint.subject_element = 'creating system'} document_usage_constraint.subject_element_value
file_name	[identification_ assignment.assigned_ id] [external_source.	41 41	33	document {document.kind -> document_type document_type.product_data_type = 'digital'}

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
	source_id]			external_identification_item = document external_identification_item <- applied_external_identification_assignment.items[1] applied_external_identification_assignment <= [external_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'file name'} identification_assignment.assigned_id] [external_identification_assignment external_identification_assignment.source -> external_source {external_source.description = 'access path'} external_source.source_id]
operating_system	document_usage_ constraint.subject_ element_value	41		document <- {document.kind -> document_type document_type.product_data_type = 'digital'} document_usage_constraint.source document_usage_constraint {document_usage_constraint.subject_element = 'operating system'} document_usage_constraint.subject_element_value
syntactical_ format	document_usage_ constraint.subject_ element_value	41		document <- {document.kind -> document_type document_type.product_data_type = 'digital'} document_usage_constraint.source document_usage_constraint {document_usage_constraint.subject_element = 'syntactical format'} document_usage_constraint.subject_element_value
DOCUMENT_ RELATIONSHIP	(document_ relationship) (product_definition_ relationship)	41 41		

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
description	(document_ relationship. description) (product_definition_ relationship. description)	41 41		
relation_type	(document_ relationship.name) (product_definition_ relationship.name)	41 41		({document_relationship (document_relationship.name) (document_relationship.name = 'copy') (document_relationship.name = 'derivation') (document_relationship.name = 'decomposition') (document_relationship.name = 'sequence') (document_relationship.name = 'substitution') (document_relationship.name = 'version association') (document_relationship.name = 'version sequence')}) ({product_definition_relationship (product_definition_relationship.name) (product_definition_relationship.name = 'copy') (product_definition_relationship.name = 'derivation') (product_definition_relationship.name = 'decomposition') (product_definition_relationship.name = 'sequence') (product_definition_relationship.name = 'substitution') (product_definition_relationship.name = 'version association') (product_definition_relationship.name = 'version sequence')})
document_ relationship to multi_language_ string (as description)	PATH		105	(document_relationship multi_language_attribute_item = document_relationship) (product_definition_relationship multi_language_attribute_item = product_definition_relationship) multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
document_ relationship to external_ document (as related)	PATH			(document_relationship document_relationship.related_document ->
document) (product_definition_relationship product_definition_relationship.related_product_definition ->
product_definition => product_definition_with_associated_documents => physically_modelled_product_definition)
document_ relationship to external_ document (as relating)	PATH			(document_relationship document_relationship.relating_document ->
document) (product_definition_relationship product_definition_relationship.relating_product_definition ->
product_definition => product_definition_with_associated_documents => physically_modelled_product_definition)
DOCUMENT_TYPE	(document_type) (drawing_definition. drawing_type)	41 101		
id	(document_type. product_data_type) (IDENTICAL MAPPING)	41		
document_type to classification_ system (as used_ classification_ system)	PATH		32	(document_type group_member_item = document_type) (drawing_definition.drawing_type drawing_definition group_member_item = drawing_definition) group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment {group_assignment.role ->
group_role group_role.name = 'classified member'} group_assignment.assigned_group ->

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
				group => (classification_system) (externally_defined_classification_system)
EXTERNAL_ DOCUMENT	(document) (physically_modelled_ product_definition)	41 214		({document document.kind -> document_type (document_type.product_data_type = 'digital') (document_type.product_data_type = 'physical')}) (physically_modelled_product_definition <= product_definition_with_associated_documents {product_definition_with_associated_documents.documentation_ids[1] -> document document.kind -> document_type document_type.product_data_type = 'physical model'})
description	document.description	41		(document {document.kind -> document_type (document_type.product_data_type = 'digital') (document_type.product_data_type = 'physical')}) (physically_modelled_product_definition <= product_definition_with_associated_documents product_definition_with_associated_documents.documentation_ids[1] -> document {document.kind -> document_type document_type.product_data_type = 'physical model'}) document.description
document_id	document.id	41		(document {document.kind -> document_type (document_type.product_data_type = 'digital') (document_type.product_data_type = 'physical')}) (physically_modelled_product_definition <= product_definition_with_associated_documents product_definition_with_associated_documents.documentation_ids[1] ->

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
				document {document.kind -> document_type document_type.product_data_type = 'physical model'}) document.id
language	document_usage_ constraint.subject_ element_value	41		(document <- {document.kind -> document_type (document_type.product_data_type = 'digital') (document_type.product_data_type = 'physical')}) (physically_modelled_product_definition <= product_definition_with_associated_documents product_definition_with_associated_documents.documentation_ids[1] -> document <- {document.kind -> document_type document_type.product_data_type = 'physical model'}) document_usage_constraint.source document_usage_constraint {document_usage_constraint.subject_element = 'language'} document_usage_constraint.subject_element_value
name	document.name	41		(document {document.kind -> document_type (document_type.product_data_type = 'digital') (document_type.product_data_type = 'physical')}) (physically_modelled_product_definition <= product_definition_with_associated_documents product_definition_with_associated_documents.documentation_ids[1] -> document {document.kind -> document_type document_type.product_data_type = 'physical model'}) document.name

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
external_ document to multi_language_ string (as description)	PATH		104	(document {document.kind -> document_type (document_type.product_data_type = 'digital') (document_type.product_data_type = 'physical')}) (physically_modelled_product_definition <= product_definition_with_associated_documents product_definition_with_associated_documents.documentation_ids[1] -> document {document.kind -> document_type document_type.product_data_type = 'physical model'}) multi_language_attribute_item = document multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
external_ document to document_type (as document_ type)	PATH			document document.kind -> document_type[1)]
external_ document to multi_language_ string (as name)	PATH		104	(document {document.kind -> document_type (document_type.product_data_type = 'digital') (document_type.product_data_type = 'physical')}) (physically_modelled_product_definition <= product_definition_with_associated_documents product_definition_with_associated_documents.documentation_ids[1] -> document

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
				{document.kind -> document_type document_type.product_data_type = 'physical model'}) multi_language_attribute_item = document multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
EXTERNAL_ DOCUMENT_ ASSIGNMENT	applied_document_ reference	214		applied_document_reference <= document_reference
role	document_role.name	41	27	applied_document_reference <= document_reference document_reference.role -> document_role document_role.name {(document_role.name) (document_role.name = 'obligatory') (document_role.name = 'informative')}
external_ document_ assignment to external_ document (as assigned_ document)	PATH			applied_document_reference <= document_reference document_reference.assigned_document -> (document {document.kind -> document_type (document_type.product_data_type = 'digital') (document_type.product_data_type = 'physical')}) (document <- {document.kind -> document_type document_type.product_data_type = 'physical model'}

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_with_associated_documents.documentation_ids[1] product_definition_with_associated_documents => physically_modelled_product_definition)
external_ document_ assignment to activity (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] ->> document_reference_item document_reference_item = action action
external_ document_ assignment to approval (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] ->> document_reference_item document_reference_item = approval approval
external_ document_ assignment to change (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] ->> document_reference_item (document_reference_item = product_definition_formation_relationship product_definition_formation_relationship) (document_reference_item = action_relationship action_relationship) (document_reference_item = configuration_item_relationship configuration_item_relationship) (document_reference_item = product_definition_relationship product_definition_relationship) (document_reference_item = shape_aspect_relationship shape_aspect_relationship)
external_ document_ assignment to class_function_ association (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] ->> document_reference_item document_reference_item = configuration_design configuration_design {configuration_design.configuration ->> configuration_item configuration_item.description = 'class function definition'}

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
external_ document_ assignment to contract (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = contract contract
external_ document_ assignment to design_ constraint (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = representation representation {representation.description = 'design constraint'}
external_ document_ assignment to design_ discipline_item_ definition (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = product_definition product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
external_ document_ assignment to drawing_sheet (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = drawing_sheet_revision drawing_sheet_revision
external_ document_ assignment to item_ classification (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = product_category product_category

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
external_ document_ assignment to item_definition_ instance_ relationship (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = product_definition_relationship product_definition_relationship {[(product_definition_relationship product_definition_relationship.related_product_definition -> product_definition => product_definition_occurrence) (product_definition_relationship => {(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')} product_definition_usage => assembly_component_usage)] [product_definition_relationship product_definition_relationship.relating_product_definition -> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template']}
external_ document_ assignment to item_definition_ relationship (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
external_ document_ assignment to item_instance (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item (document_reference_item = product_definition_occurrence product_definition_occurrence) (document_reference_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})
external_ document_ assignment to item_instance_ relationship (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item (document_reference_item = product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}) (document_reference_item = product_definition_substitute product_definition_substitute)
external_ document_ assignment to item_solution (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = solution_based_product_definition solution_based_product_definition

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
external_ document_ assignment to material (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = material_designation material_designation
external_ document_ assignment to physical_ instance (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}
external_ document_ assignment to physical_ instance_usage (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'physical occurrence usage'}
external_ document_ assignment to process_ operation (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = action action {action.description = 'process operation'}

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
external_ document_ assignment to process_ operation_ resource (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = resource_requirement_type resource_requirement_type
external_ document_ assignment to process_plan (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = process_plan process_plan
external_ document_ assignment to product_class (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = product_class product_class
external_ document_ assignment to product_ component (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = product_component product_component
external_ document_ assignment to product_function (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition']}

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
				[product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}
external_ document_ assignment to product_ specification (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = configurable_item configurable_item {configurable_item <= configuration_item configuration_item.description = 'product specification'}
external_ document_ assignment to project (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = project project
external_ document_ assignment to property (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item (document_reference_item = property_definition property_definition) (document_reference_item = action_property action_property)
external_ document_ assignment to retention_period (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = retention retention

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
external_ document_ assignment to specification (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item (document_reference_item = product_concept_feature product_concept_feature) (document_reference_item = package_product_concept_feature package_product_concept_feature)
external_ document_ assignment to specification_ category (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item (document_reference_item = product_concept_feature_category product_concept_feature_category) (document_reference_item = exclusive_product_concept_feature_category exclusive_product_concept_feature_category)
external_ document_ assignment to surface_ condition (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = representation representation {[(representation.description = 'coating layer') (representation.description = 'contact ratio') (representation.description = 'hardness') (representation.description = 'roughness') (representation.description = 'tactile appearance') (representation.description = 'treatment result') (representation.description = 'visual appearance')] [representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition => surface_condition]}

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
external_ document_ assignment to view_area (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item (document_reference_item = presentation_area presentation_area) (document_reference_item = mechanical_design_geometric_presentation_area mechanical_design_geometric_presentation_area)
external_ document_ assignment to work_order (as is_assigned_ to)	PATH			applied_document_reference applied_document_reference.items[i] -> document_reference_item document_reference_item = action_directive action_directive
EXTERNAL_ DOCUMENT_VERSION	document	41	170,33	{document version_item = document version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
version_id	identification_ assignment.assigned_ id	41	170,33	document version_item = document version_item <- version_assignment.items[1] version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
EXTERNAL_LIBRARY_ REFERENCE	known_source	214		known_source <= [external_source] [pre_defined_item {pre_defined_item.name = 'ISO 13584 dictionary'}]

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
external_id	external_source. source_id	41		known_source <= external_source external_source.source_id {external_source.source_id -> source_item source_item = identifier identifier}
EXTERNAL_MODEL	externally_defined_ representation	214	43,71	externally_defined_representation <= representation
description	representation. description	43		externally_defined_representation <= representation representation.description
model_id	representation.id	43		externally_defined_representation <= representation representation.id
role	representation.name	43		externally_defined_representation <= representation representation.name {(representation.name) (representation.name = 'design shape') (representation.name = 'idealized shape')}
external_model to multi_ language_string (as description)	PATH		114	externally_defined_representation multi_language_attribute_item = externally_defined_representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
external_model to digital_ document (as is_defined_ as)	PATH		27	externally_defined_representation digital_document_reference_item = externally_defined_representation digital_document_reference_item <- digital_document_reference.items[1] digital_document_reference <= document_reference {document_reference.role ->

Table 5 – Mapping table external_reference_mechanism UoF (E1) (continued)

Application element	AIM element	Source	Rules	Reference path
				document_role document_role.name = 'external definition'} document_reference.assigned_document -> document {document.kind -> document_type document_type.product_data_type = 'digital'}
external_model to cartesian_ coordinate_space (as is_defined_ in)	PATH			externally_defined_representation <= representation representation.context_of_items -> representation_context => geometric_representation_context
PHYSICAL_MODEL	physically_modelled_ product_definition	214	198	physically_modelled_product_definition <= product_definition_with_associated_documents {[product_definition_with_associated_documents.documentation_ids[1] -> document document.kind -> document_type document_type.product_data_type = 'physical model'] [product_definition_with_associated_documents <= product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence']}
found_in	organizational_ address	41		physically_modelled_product_definition organization_item = physically_modelled_product_definition organization_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment.role -> organization_role organization_role.name = 'location'} organization_assignment.assigned_organization -> organization <-

Table 5 – Mapping table external_reference_mechanism UoF (E1) (concluded)

Application element	AIM element	Source	Rules	Reference path
				organizational_address.organizations[i] organizational_address
scale	measure_ representation_item	45		physically_modelled_product_definition <= product_definition_with_associated_documents <= product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[1] -> representation_item => {representation_item.name = 'scale'} measure_representation_item {measure_representation_item <= measure_with_unit => ratio_measure_with_unit}

NOTE – The following note numbers correspond to the footnotemarks listed in the table above:

1 Collision with document_type.product_data_type = 'digital' or 'physical' or 'physical_model' has to be resolved by appropriate instantiation practices.

Table 6 – Mapping table user_defined_feature UoF (FF1)

Application element	AIM element	Source	Rules	Reference path
FEATURE_ OCCURRENCE	shape_aspect	41		
feature_function	shape_aspect. description	41		1)
feature_ occurrence to multi_language_ string (as feature_ function)	PATH		159	shape_aspect multi_language_attribute_item = shape_aspect multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
FEATURE_ PARAMETER	measure_ representation_item	45		
parameter_name	identification_ assignment.assigned_ id	41		measure_representation_item identification_item = measure_representation_item identification_item <- applied_identification_assignment.items[1] applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'parameter id'} identification_assignment.assigned_id
feature_ parameter to value_with_unit (as specified_ value)	PATH			measure_representation_item <= measure_with_unit
GENERAL_COMPOUND_ FEATURE	composite_shape_ aspect	47		{composite_shape_aspect <= shape_aspect shape_aspect.product_definitional = 'TRUE'}

Table 6 – Mapping table user_defined_feature UoF (FF1) (continued)

Application element	AIM element	Source	Rules	Reference path
general_compound_ feature to feature_ definition (as elements)	PATH			composite_shape_aspect composite_shape_aspect.component_relationships[i] -> shape_aspect_relationship {shape_aspect_relationship => feature_component_relationship} shape_aspect_relationship.related_shape_aspect -> shape_aspect => instanced_feature <= feature_definition
general_compound_ feature to general_feature (as elements)	PATH			composite_shape_aspect composite_shape_aspect.component_relationships[i] -> shape_aspect_relationship {shape_aspect_relationship => feature_component_relationship} shape_aspect_relationship.related_shape_aspect -> shape_aspect {group_member_item = shape_aspect group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment group_assignment.role -> group_role group_role.name = 'feature categorization'}
general_compound_ feature to thread_feature (as elements)	PATH			composite_shape_aspect composite_shape_aspect.component_relationships[i] -> shape_aspect_relationship {shape_aspect_relationship => feature_component_relationship} shape_aspect_relationship.related_shape_aspect -> shape_aspect => thread_occurrence

Table 6 – Mapping table user_defined_feature UoF (FF1) (continued)

Application element	AIM element	Source	Rules	Reference path
general_compound_ feature to transition_ feature (as elements)	PATH			composite_shape_aspect composite_shape_aspect.component_relationships[i] -> shape_aspect_relationship {shape_aspect_relationship => feature_component_relationship} shape_aspect_relationship.related_shape_aspect -> shape_aspect => transition_feature
GENERAL_FEATURE	shape_aspect	41	32	{shape_aspect [shape_aspect.product_definitional = 'TRUE'] [group_member_item = shape_aspect group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment group_assignment.role -> group_role group_role.name = 'feature categorization']}
feature_category	group.description	41		shape_aspect group_member_item = shape_aspect group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment {group_assignment.role -> group_role group_role.name = 'feature categorization'} group_assignment.assigned_group -> group group.description
feature_type	group.name	41		shape_aspect group_member_item = shape_aspect group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment

Table 6 – Mapping table user_defined_feature UoF (FF1) (continued)

Application element	AIM element	Source	Rules	Reference path
				{group_assignment.role -> group_role group_role.name = 'feature categorization'} group_assignment.assigned_group -> group group.name
general_feature to general_ feature_ parameter (as feature_ parameter)	PATH			shape_aspect {[shape_aspect.product_definitional = 'TRUE'] [group_member_item = shape_aspect group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment group_assignment.role -> group_role group_role.name = 'feature categorization']} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item => measure_representation_item
general_feature to axis_ placement (as location_and_ orientation)	PATH			shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation}

Table 6 – Mapping table user_defined_feature UoF (FF1) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item => {representation_item.name = 'location and orientation'} geometric_representation_item => placement
GENERAL_FEATURE_ PARAMETER	measure_ representation_item	45		
parameter_role	representation_item. name	43		measure_representation_item <= representation_item representation_item.name
general_feature_ parameter to multi_language_ string (as parameter_ role)	PATH		120	measure_representation_item multi_language_attribute_item = measure_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation_item.name'}
general_feature_ parameter to feature_ parameter (as parameter_ value)	IDENTICAL MAPPING			
INCLUDED_FEATURE	(shape_aspect) (instanced_feature)	41 214		({(shape_aspect group_member_item = shape_aspect group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment group_assignment.role ->

Table 6 – Mapping table user_defined_feature UoF (FF1) (continued)

Application element	AIM element	Source	Rules	Reference path
				group_role group_role.name = 'feature categorization') (shape_aspect => thread_occurrence) (shape_aspect => transition_feature)}) (instanced_feature <= [shape_aspect] [feature_definition <= characterized_object])
included_feature to feature_ definition (as definition)	PATH			instanced_feature <= feature_definition
included_feature to general_ feature (as definition)	IDENTICAL MAPPING			{shape_aspect group_member_item = shape_aspect group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment group_assignment.role -> group_role group_role.name = 'feature categorization'}
included_feature to thread_ feature (as definition)	PATH			shape_aspect => thread_occurrence
included_feature to transition_ feature (as definition)	PATH			shape_aspect => transition_feature
VALUE_WITH_UNIT	measure_with_unit	41	35	
unit_component	(named_unit) (derived_unit)	41 41	25	measure_with_unit measure_with_unit.unit_component -> (unit = named_unit

Table 6 – Mapping table user_defined_feature UoF (FF1) (concluded)

Application element	AIM element	Source	Rules	Reference path
				named_unit) (unit = derived_unit derived_unit)
value_component	measure_value	41		measure_with_unit measure_with_unit.value_component ->
measure_value

NOTE – The following note numbers correspond to the footnotemarks listed in the table above:

1 In case the attributes feature_occurrence.feature_function and shape_aspect.description are both instantiated in the ARM, they have to be combined into one attribute shape_aspect.description in the AIM.

Table 7 – Mapping table included_feature UoF (FF2)

Application element	AIM element	Source	Rules	Reference path
ANGLE_ SPECIFICATION	directed_angle	214		directed_angle <= shape_aspect {shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition}
angle_reference	descriptive_ representation_item	45		directed_angle <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'angle reference'} descriptive_representation_item {(descriptive_representation_item.description = 'punch direction') (descriptive_representation_item.description = 'profile normal') (descriptive_representation_item.description = 'surface normal')}
angle_ specification to feature_ parameter (as angle)	PATH			directed_angle <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
ANGLE_TAPER	taper	214		taper <= shape_aspect {[shape_aspect.description = 'angle taper'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
angle_taper to feature_ parameter (as angle)	PATH			taper <= shape_aspect {shape_aspect.description = 'angle taper'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{representation => shape_representation => shape_representation_with_parameters} representation.items[i] ->> representation_item => {representation_item.name = 'taper angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
BEAD_END_TYPE	bead_end	214		bead_end <= shape_aspect {[(shape_aspect.description = 'linear') (shape_aspect.description = 'open') (shape_aspect.description = 'radiused')] [shape_aspect.of_shape ->> product_definition_shape <= property_definition property_definition.definition ->> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
is_applied_to	shape_aspect.name	41		bead_end <= shape_aspect <- {(shape_aspect.description = 'linear') (shape_aspect.description = 'open') (shape_aspect.description = 'radiused')} shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship {shape_aspect_relationship.name = 'bead end usage'} shape_aspect_relationship.related_shape_aspect ->> shape_aspect {shape_aspect.description = 'bead end occurrence'} shape_aspect.name {(shape_aspect.name = 'end condition at path start')

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				(shape_aspect.name = 'end condition at path end')}
BEAD_FEATURE	bead	214		bead <= feature_definition <= characterized_object
maximum_depth	measure_ representation_item	45		bead <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'maximum depth'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
bead_feature to feature_ parameter (as base_radius)	PATH			bead <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <-

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'base radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
bead_feature to open_profile (as bead_ profile)	PATH			bead <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'bead profile'] [shape_aspect.description = 'open profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{[shape_aspect_relationship.name = 'open profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => (open_path_profile) (partial_circular_profile) (rounded_u_profile) (square_u_profile) (tee_profile) (vee_profile)
bead_feature to bead_end_type (as end_ condition)	PATH			bead <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[(shape_aspect.name = 'end condition at path start') (shape_aspect.name = 'end condition at path end')] [shape_aspect.description = 'bead end occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'bead end usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {(shape_aspect.description = 'linear') (shape_aspect.description = 'open') (shape_aspect.description = 'radiused')} bead_end

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
bead_feature to transformation (as profile_ placement)	PATH			bead <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape [shape_aspect <- {[shape_aspect.name = 'bead profile'] [shape_aspect.description = 'open profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'open profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect {shape_aspect => (open_path_profile) (partial_circular_profile) (rounded_u_profile) (square_u_profile) (tee_profile) (vee_profile)} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation => shape_representation_with_parameters} representation.items[i] ->> representation_item <- {[representation_item.name = 'orientation'] [representation_item => geometric_representation_item => placement]} representation_map.mapping_origin] [represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->> representation {representation => shape_representation} representation.items[i] ->> representation_item <- {[representation_item.name = 'profile placement'] [representation_item => geometric_representation_item => placement]} mapped_item.mapping_target]
bead_feature to path (as sweep_path)	PATH			bead <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect <- {[shape_aspect.name = 'sweep path'] [shape_aspect.description = 'path feature component occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'path feature component usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => path_feature_component
BOSS_FEATURE	boss	214		boss <= feature_definition <= characterized_object {(characterized_object.description = 'circular') (characterized_object.description = 'rectangular') (characterized_object.description = 'general')}
boss_feature to feature_ parameter (as fillet_ radius)	PATH			boss <= feature_definition <= characterized_object {(characterized_object.description = 'circular') (characterized_object.description = 'rectangular') (characterized_object.description = 'general')} characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'fillet radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
boss_feature to transformation (as profile_ placement)	PATH			boss <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape [shape_aspect <- {[shape_aspect.name = 'boundary'] [shape_aspect.description = 'closed profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'closed profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect {shape_aspect => (circular_closed_profile) (rectangular_closed_profile) (closed_path_profile)} represented_definition = shape_aspect

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item <- {[representation_item.name = 'orientation'] [representation_item => geometric_representation_item => placement]} representation_map.mapping_origin] [represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item <- {[representation_item.name = 'profile placement'] [representation_item => geometric_representation_item => placement]} mapped_item.mapping_target]

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
boss_feature to linear_path (as sweep_path)	PATH			boss <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'sweep path'] [shape_aspect.description = 'path feature component occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'path feature component usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'linear'} path_feature_component
boss_feature to angle_taper (as taper)	PATH			boss <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'taper'] [shape_aspect.description = 'taper occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'taper usage']

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				[shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'angle taper'} taper
boss_feature to diameter_taper (as taper)	PATH			boss <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'taper'] [shape_aspect.description = 'taper occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'taper usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'diameter taper'} taper
boss_feature to directed_taper (as taper)	PATH			boss <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect <- {[shape_aspect.name = 'taper'] [shape_aspect.description = 'taper occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'taper usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'directed taper'} taper
boss_feature to boss_top_ condition (as top_ condition)	PATH			boss <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'top condition'] [shape_aspect.description = 'boss top occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'boss top usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {(shape_aspect.description = 'planar') (shape_aspect.description = 'non-planar')} boss_top

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
BOSS_TOP_ CONDITION	boss_top	214		boss_top <= shape_aspect {[shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition] [(shape_aspect.description = 'planar') (shape_aspect.description = 'non-planar')]}
is_applied_to	descriptive_ representation_item	45		boss_top <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'top location'} descriptive_representation_item {(descriptive_representation_item.description = 'path start') (descriptive_representation_item.description = 'path end')}
CATALOG_THREAD	externally_defined_ thread	214		externally_defined_thread <= [feature_definition <= characterized_object] [externally_defined_item]

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
catalog_thread to external_ document (as is_specified_ in)	PATH		27	externally_defined_thread document_reference_item = externally_defined_thread document_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference {document_reference.role -> document_role document_role.name = 'specification'} document_reference.assigned_document -> document
catalog_thread to external_ library_ reference (as is_specified_ in)	PATH			externally_defined_thread <= externally_defined_item externally_defined_item.source -> external_source => known_source
CHAMFER	chamfer	214		chamfer <= transition_feature <= shape_aspect
chamfer to chamfer_ definition (as is_defined_ by)	PATH			chamfer <= transition_feature <= shape_aspect <- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship => {[(shape_aspect_relationship.name = 'first chamfer offset') (shape_aspect_relationship.name = 'second chamfer offset')] [shape_aspect_relationship.related_shape_aspect -> shape_aspect => chamfer_offset]} feature_component_relationship
CHAMFER_ANGLE	chamfer_offset	214		chamfer_offset <= shape_aspect {shape_aspect.description = 'chamfer angle'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
chamfer_angle to feature_ parameter (as angle)	PATH			chamfer_offset <= shape_aspect {shape_aspect.description = 'chamfer angle'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'offset angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
CHAMFER_ DEFINITION	feature_component_ relationship	214		feature_component_relationship <= shape_aspect_relationship {[(shape_aspect_relationship.name = 'first chamfer offset') (shape_aspect_relationship.name = 'second chamfer offset')] [shape_aspect_relationship.related_shape_aspect -> shape_aspect => chamfer_offset]}
chamfer_ definition to chamfer_offset (as offset_1)	PATH			feature_component_relationship <= shape_aspect_relationship {shape_aspect_relationship.name = 'first chamfer offset'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => {shape_aspect.description = 'chamfer offset'} chamfer_offset

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
chamfer_ definition to chamfer_angle (as value_2)	PATH			feature_component_relationship <= shape_aspect_relationship {shape_aspect_relationship.name = 'second chamfer offset'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => {shape_aspect.description = 'chamfer angle'} chamfer_offset
chamfer_ definition to chamfer_offset (as value_2)	PATH			feature_component_relationship <= shape_aspect_relationship {shape_aspect_relationship.name = 'second chamfer offset'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => {shape_aspect.description = 'chamfer offset'} chamfer_offset
CHAMFER_OFFSET	chamfer_offset	214		chamfer_offset <= shape_aspect {shape_aspect.description = 'chamfer offset'}
chamfer_offset to feature_ parameter (as offset)	PATH			chamfer_offset <= shape_aspect {shape_aspect.description = 'chamfer offset'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'offset amount'} measure_representation_item {measure_representation_item <=

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				measure_with_unit => length_measure_with_unit}
CIRCULAR_BOSS	boss	214		boss <= feature_definition <= characterized_object {characterized_object.description = 'circular'}
circular_boss to circular_closed_ profile (as boundary)	PATH			boss <= feature_definition <= characterized_object {characterized_object.description = 'circular'} characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'boundary'] [shape_aspect.description = 'closed profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'closed profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => circular_closed_profile
CIRCULAR_CLOSED_ PROFILE	circular_closed_ profile	214		circular_closed_profile <= shape_aspect {shape_aspect.of_shape ->} product_definition_shape <= property_definition property_definition.definition ->} characterized_definition characterized_definition = characterized_object characterized_object =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				feature_component_definition}
circular_closed_ profile to feature_ parameter (as diameter)	PATH			circular_closed_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'diameter'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
CIRCULAR_OFFSET_ PATTERN	shape_aspect	41		{shape_aspect [shape_aspect.description = 'circular offset'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object <- {characterized_object => feature_definition => replicate_feature => modified_pattern} characterized_object_relationship.related_object characterized_object_relationship

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{characterized_object_relationship.name = 'replicate base assignment'} characterized_object_relationship.relating_object->characterized_object => feature_definition => replicate_feature => circular_pattern]}
angular_offset	measure_representation_item	45		shape_aspect {shape_aspect.description = 'circular offset'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'angular offset'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
index	measure_representation_item	45		shape_aspect {shape_aspect.description = 'circular offset'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{representation => shape_representation => shape_representation_with_parameters} representation.items[i] ->> representation_item => {representation_item.name = 'offset index'} measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component ->> measure_value measure_value = count_measure count_measure}
CIRCULAR_OMIT_ PATTERN	shape_aspect	41		{shape_aspect [shape_aspect.description = 'circular omit'] [shape_aspect.of_shape ->> product_definition_shape <= property_definition property_definition.definition ->> characterized_definition characterized_definition = characterized_object characterized_object <- {characterized_object => feature_definition => replicate_feature => modified_pattern} characterized_object_relationship.related_object characterized_object_relationship {characterized_object_relationship.name = 'replicate base assignment'} characterized_object_relationship.relating_object->> characterized_object => feature_definition => replicate_feature => circular_pattern]}
index	measure_ representation_item	45		shape_aspect {shape_aspect.description = 'circular omit'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'omit index'} measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure}
CIRCULAR_PATH	path_feature_ component	214		path_feature_component <= shape_aspect {[(shape_aspect.description = 'partial circular') (shape_aspect.description = 'complete circular')] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
circular_path to feature_ parameter (as radius)	PATH			path_feature_component <= shape_aspect {(shape_aspect.description = 'partial circular') (shape_aspect.description = 'complete circular')} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
circular_path to axis_placement (as rotation_ axis_defined_by)	PATH			path_feature_component <= shape_aspect {(shape_aspect.description = 'partial circular') (shape_aspect.description = 'complete circular')} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'orientation'} geometric_representation_item => placement
CIRCULAR_PATTERN	circular_pattern	214		circular_pattern <= replicate_feature <= feature_definition <= characterized_object
number_of_ features	measure_ representation_item	45		circular_pattern <= replicate_feature <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'number of features'} measure_representation_item {measure_representation_item <=

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure}
circular_pattern to feature_ parameter (as angular_ spacing)	PATH			circular_pattern <= replicate_feature <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'angular spacing'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
circular_pattern to feature_ parameter (as diameter)	PATH			circular_pattern <= replicate_feature <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'diameter'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
circular_pattern to circular_omit_ pattern (as missing_ feature)	PATH			circular_pattern <= replicate_feature <= feature_definition <= characterized_object <- characterized_object_relationship.relating_object characterized_object_relationship {characterized_object_relationship.name = 'replicate base assignment'} characterized_object_relationship.related_object ->

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_object {characterized_object => feature_definition => replicate_feature => modified_pattern} characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect {shape_aspect.description = 'circular omit'}
circular_pattern to circular_ offset_pattern (as relocated_ feature)	PATH			circular_pattern <= replicate_feature <= feature_definition <= characterized_object <- characterized_object_relationship.relating_object characterized_object_relationship {characterized_object_relationship.name = 'replicate base assignment'} characterized_object_relationship.related_object -> characterized_object {characterized_object => feature_definition => replicate_feature => modified_pattern} characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect {shape_aspect.description = 'circular offset'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
circular_pattern to feature_ parameter (as rotation)	PATH			circular_pattern <= replicate_feature <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'base feature rotation'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
CLOSED_PROFILE	(circular_closed_ profile) (ngon_closed_ profile) (rectangular_closed_ profile) (closed_path_ profile)	214 214 214 214		(circular_closed_profile <=) (ngon_closed_profile <=) (rectangular_closed_profile <=) (closed_path_profile <=) shape_aspect {shape_aspect.of_shape -> product_definition_shape <= property_definition

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition}
CLOSED_PROFILE_ USAGE	shape_defining_ relationship	214		shape_defining_relationship <= shape_aspect_relationship {[shape_aspect_relationship.name = 'closed profile usage'] [shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'closed profile occurrence'} shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_definition => (hole_in_panel) (feature_in_panel)]}
closed_profile_ usage to transformation (as placement)	PATH			shape_defining_relationship <= shape_aspect_relationship {shape_aspect_relationship.name = 'closed profile usage'} [shape_aspect_relationship.relating_shape_aspect -> shape_aspect {shape_aspect => (circular_closed_profile) (ngon_closed_profile) (rectangular_closed_profile) (closed_path_profile)} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item <- {[representation_item.name = 'orientation'] [representation_item => geometric_representation_item => placement]} representation_map.mapping_origin] [shape_aspect_relationship.related_shape_aspect -> shape_aspect {[shape_aspect.description = 'closed profile occurrence'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_definition => (hole_in_panel) (feature_in_panel)]} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{representation => shape_representation} representation.items[i] -> representation_item <- {[representation_item.name = 'profile placement'] [representation_item => geometric_representation_item => placement]} mapped_item.mapping_target]
closed_profile_ usage to closed_ profile (as profile)	PATH			shape_defining_relationship <= shape_aspect_relationship {shape_aspect_relationship.name = 'closed profile usage'} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => (circular_closed_profile) (ngon_closed_profile) (rectangular_closed_profile) (closed_path_profile)
COMPLETE_ CIRCULAR_PATH	path_feature_ component	214		path_feature_component <= shape_aspect {[shape_aspect.description = 'complete circular'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
COMPOUND_FEATURE_ IN_PANEL	compound_feature_in_ panel	214		compound_feature_in_panel <= feature_definition <= characterized_object
feature_type	characterized_object. description	41		compound_feature_in_panel <= feature_definition <= characterized_object characterized_object.description

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
COMPOUND_FEATURE_ IN_PANEL_ELEMENT	(instanced_feature) (placed_feature) (transition_feature)	214 214 214		(instanced_feature <= [shape_aspect] [feature_definition <= characterized_object]) (placed_feature <= shape_aspect) (transition_feature <= shape_aspect)
compound_feature_ in_panel_element to transformation (as element_ placement)	PATH			placed_feature <= shape_aspect {shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship <- {shape_aspect_relationship => feature_component_relationship} composite_shape_aspect.component_relationships[i] composite_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_definition => compound_feature_in_panel} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation} [representation.items[i] -> representation_item <- {[representation_item.name = 'element placement'] [representation_item => geometric_representation_item => placement]} mapped_item.mapping_target] [representation.items[i] -> representation_item => mapped_item mapped_item.mapping_source -> representation_map {representation_map.mapped_representation -> representation <- {representation => shape_representation => shape_representation_with_parameters} property_definition_representation.used_representation property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_definition =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				(bead) (hole_in_panel) (feature_in_panel) (joggle) (locator) (compound_feature_in_panel)} representation_map.mapping_origin]
compound_feature_ in_panel_element to process_ feature_in_panel (as feature)	PATH			(instanced_feature <= feature_definition => (bead) (hole_in_panel) (feature_in_panel) (joggle) (locator) (compound_feature_in_panel)) (placed_feature <= shape_aspect {shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship <- {shape_aspect_relationship => feature_component_relationship} composite_shape_aspect.component_relationships[i] composite_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_definition => compound_feature_in_panel}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item => mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- {representation => shape_representation => shape_representation_with_parameters} property_definition_representation.used_representation property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_definition => (bead)

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				(hole_in_panel) (feature_in_panel) (joggle) (locator) (compound_feature_in_panel))
compound_feature_ in_panel_element to transition_ feature (as feature)	IDENTICAL MAPPING			
compound_feature_ in_panel_element to compound_ feature_in_panel (as is_element_ for)	PATH			((instanced_feature <=) (transition_feature <=) shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship <- {shape_aspect_relationship => feature_component_relationship} composite_shape_aspect.component_relationships[i] composite_shape_aspect <= shape_aspect => instanced_feature <=) ((placed_feature <=) (transition_feature <=) shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship <- {shape_aspect_relationship => feature_component_relationship} composite_shape_aspect.component_relationships[i] composite_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition ->

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition characterized_definition = characterized_object characterized_object =>) feature_definition => compound_feature_in_panel
COMPOUND_FEATURE_IN_PANEL_ELEMENT_RELATIONSHIP	shape_aspect_relationship	41		{shape_aspect_relationship shape_aspect_relationship.name = 'compound feature ordering'}
compound_feature_in_panel_element_relationship to compound_feature_in_panel_element (as predecessor)	PATH			shape_aspect_relationship {shape_aspect_relationship.name = 'compound feature ordering'} shape_aspect_relationship.relating_shape_aspect ->shape_aspect => (instanced_feature) (placed_feature) (transition_feature)
compound_feature_in_panel_element_relationship to compound_feature_in_panel_element (as successor)	PATH			shape_aspect_relationship {shape_aspect_relationship.name = 'compound feature ordering'} shape_aspect_relationship.related_shape_aspect ->shape_aspect => (instanced_feature) (placed_feature) (transition_feature)
COMPOUND_FEATURE_IN_SOLID	compound_feature_in_solid	214		compound_feature_in_solid <= feature_definition <= characterized_object
feature_type	characterized_object.description	41		compound_feature_in_solid <= feature_definition <= characterized_object characterized_object.description
COMPOUND_FEATURE_IN_SOLID_ELEMENT	(instanced_feature) (placed_feature) (transition_feature) (thread_occurrence)	214 214 214 214		(instanced_feature <= [shape_aspect] [feature_definition <= characterized_object]) (placed_feature <= shape_aspect)

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				(transition_feature <= shape_aspect) (thread_occurrence <= shape_aspect)
compound_feature_ in_solid_element to transformation (as element_ placement)	PATH			placed_feature <= shape_aspect {shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship <- {shape_aspect_relationship => feature_component_relationship} composite_shape_aspect.component_relationships[i] composite_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_definition => compound_feature_in_solid} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} [representation.items[i] -> representation_item <-

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{[representation_item.name = 'element placement'] [representation_item => geometric_representation_item => placement]} mapped_item.mapping_target] [representation.items[i] -> representation_item => mapped_item mapped_item.mapping_source -> representation_map {representation_map.mapped_representation -> representation <- {representation => shape_representation => shape_representation_with_parameters} property_definition_representation.used_representation property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_definition => (slot) (pocket) (boss) (round_hole)

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				(rib) (compound_feature_in_solid)} representation_map.mapping_origin]
compound_feature_ in_solid_element to process_ feature_in_solid (as feature)	PATH			(instanced_feature <= feature_definition => (slot) (pocket) (boss) (round_hole) (rib) (compound_feature_in_solid)) (placed_feature <= shape_aspect {shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship <- {shape_aspect_relationship => feature_component_relationship} composite_shape_aspect.component_relationships[i] composite_shape_aspect <= shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_definition => compound_feature_in_solid} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item => mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- {representation => shape_representation => shape_representation_with_parameters} property_definition_representation.used_representation property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_definition => (slot) (pocket) (boss) (round_hole) (rib)

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				(compound_feature_in_solid))
compound_feature_ in_solid_element to thread_ feature (as feature)	IDENTICAL MAPPING			
compound_feature_ in_solid_element to transition_ feature (as feature)	IDENTICAL MAPPING			
compound_feature_ in_solid_element to compound_ feature_in_solid (as is_element_ for)	PATH			((instanced_feature <=) (transition_feature <=) (thread_occurrence <=) shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship <- {shape_aspect_relationship => feature_component_relationship} composite_shape_aspect.component_relationships[i] composite_shape_aspect <= shape_aspect => instanced_feature <=) ((placed_feature <=) (transition_feature <=) (thread_occurrence <=) shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship <- {shape_aspect_relationship => feature_component_relationship} composite_shape_aspect.component_relationships[i] composite_shape_aspect <= shape_aspect shape_aspect.of_shape ->

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object =>) feature_definition => compound_feature_in_solid
COMPOUND_FEATURE_IN_SOLID_ELEMENT_RELATIONSHIP	shape_aspect_relationship	41		{shape_aspect_relationship shape_aspect_relationship.name = 'compound feature ordering'}
compound_feature_in_solid_element_relationship to compound_feature_in_solid_element (as predecessor)	PATH			shape_aspect_relationship {shape_aspect_relationship.name = 'compound feature ordering'} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => (instanced_feature) (placed_feature) (transition_feature) (thread_occurrence)
compound_feature_in_solid_element_relationship to compound_feature_in_solid_element (as successor)	PATH			shape_aspect_relationship {shape_aspect_relationship.name = 'compound feature ordering'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => (instanced_feature) (placed_feature) (transition_feature) (thread_occurrence)
CONICAL_BOTTOM	hole_bottom	214		hole_bottom <= shape_aspect {[shape_aspect.description = 'conical'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_object characterized_object => feature_component_definition]}
conical_bottom to feature_ parameter (as tip_angle)	PATH			hole_bottom <= shape_aspect {shape_aspect.description = 'conical'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'tip angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
conical_bottom to feature_ parameter (as tip_radius)	PATH			hole_bottom <= shape_aspect {shape_aspect.description = 'conical'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation => shape_representation_with_parameters} representation.items[i] ->> representation_item => {representation_item.name = 'tip radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
CONSTANT_RADIUS_ EDGE_ROUND	edge_round	214		edge_round <= transition_feature <= shape_aspect {shape_aspect.description = 'constant radius'}
constant_radius_ edge_round to feature_ parameter (as first_ offset)	PATH			edge_round <= transition_feature <= shape_aspect {shape_aspect.description = 'constant radius'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] ->> representation_item => {representation_item.name = 'first face offset'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
constant_radius_ edge_round to feature_ parameter (as radius)	PATH			edge_round <= transition_feature <= shape_aspect {shape_aspect.description = 'constant radius'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
constant_radius_ edge_round to feature_ parameter (as second_ offset)	PATH			edge_round <= transition_feature <= shape_aspect {shape_aspect.description = 'constant radius'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'second face offset'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
CONSTANT_RADIUS_ FILLET	fillet	214		fillet <= transition_feature <= shape_aspect {shape_aspect.description = 'constant radius'}
constant_radius_ fillet to feature_ parameter (as first_ offset)	PATH			fillet <= transition_feature <= shape_aspect {shape_aspect.description = 'constant radius'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'first face offset'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
constant_radius_ fillet to feature_ parameter (as radius)	PATH			fillet <= transition_feature <= shape_aspect {shape_aspect.description = 'constant radius'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
constant_radius_ fillet to feature_ parameter (as second_ offset)	PATH			fillet <= transition_feature <= shape_aspect {shape_aspect.description = 'constant radius'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'second face offset'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
DEFAULT_CHAMFER	representation	43		{representation [representation.description = 'default chamfer'] [representation.context_of_items ->] representation_context representation_context.context_type = 'default setting']}
description	descriptive_ representation_item	45		representation {representation.description = 'default chamfer'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item
default_chamfer to multi_ language_string (as description)	PATH		101	representation {representation.description = 'default chamfer'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
default_chamfer to feature_parameter (as offset_value)	PATH			representation {representation.description = 'default chamfer'} representation.items[i] -> representation_item => {representation_item.name = 'offset value'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
DEFAULT_DRAFT_ANGLE	representation	43		{representation [representation.description = 'default draft angle'] [representation.context_of_items -> representation_context representation_context.context_type = 'default setting']}
description	descriptive_representation_item	45		representation {representation.description = 'default draft angle'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item
default_draft_angle to multi_language_string (as description)	PATH		101	representation {representation.description = 'default draft angle'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
default_draft_ angle to feature_ parameter (as draft_angle_ value)	PATH			representation {representation.description = 'default draft angle'} representation.items[i] -> representation_item => {representation_item.name = 'draft angle value'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
DEFAULT_EDGE_ ROUND	representation	43		{representation [representation.description = 'default edge round'] [representation.context_of_items -> representation_context representation_context.context_type = 'default setting']}
description	descriptive_ representation_item	45		representation {representation.description = 'default edge round'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item
default_edge_ round to multi_ language_string (as description)	PATH		101	representation {representation.description = 'default edge round'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
default_edge_ round to feature_ parameter (as radius_ value)	PATH			representation {representation.description = 'default edge round'} representation.items[i] -> representation_item => {representation_item.name = 'radius value'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
DEFAULT_FILLET	representation	43		{representation [representation.description = 'default fillet'] [representation.context_of_items ->] representation_context representation_context.context_type = 'default setting']}
description	descriptive_ representation_item	45		representation {representation.description = 'default fillet'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item
default_fillet to multi_ language_string (as description)	PATH		101	representation {representation.description = 'default fillet'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
default_fillet to feature_parameter (as radius_value)	PATH			representation {representation.description = 'default fillet'} representation.items[i] -> representation_item => {representation_item.name = 'radius value'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
DEFAULT_SETTING_ASSOCIATION	(property_definition_representation) (representation_relationship)	41 43		({property_definition_representation property_definition_representation.name = 'default setting'}) ({representation_relationship representation_relationship.name = 'default setting'})
default_setting_association to digital_document (as default_context)	PATH		27	representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_2 -> representation => externally_defined_representation digital_document_reference_item = externally_defined_representation digital_document_reference_item <- digital_document_reference.items[1] digital_document_reference <= document_reference {document_reference.role -> document_role document_role.name = 'external definition'} document_reference.assigned_document -> document {document.kind -> document_type document_type.product_data_type = 'digital'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
default_setting_ association to drawing_sheet (as default_ context)	PATH			representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_2 ->>>>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
default_setting_ association to default_draft_ angle (as default_ value)	PATH			(property_definition_representation {property_definition_representation.name = 'default setting'} property_definition_representation.used_representation ->) (representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_1 ->) representation {representation.description = 'default draft angle'}
default_setting_ association to default_edge_ round (as default_ value)	PATH			(property_definition_representation {property_definition_representation.name = 'default setting'} property_definition_representation.used_representation ->) (representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_1 ->) representation {representation.description = 'default edge round'}
default_setting_ association to default_fillet (as default_ value)	PATH			(property_definition_representation {property_definition_representation.name = 'default setting'} property_definition_representation.used_representation ->) (representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_1 ->) representation {representation.description = 'default fillet'}
DEFINED_THREAD	thread	214		thread <= feature_definition <= characterized_object
fit_class	descriptive_ representation_item	45		thread <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'fit class'} descriptive_representation_item
form	descriptive_ representation_item	45		thread <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] ->

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item => {representation_item.name = 'form'} descriptive_representation_item
hand	descriptive_ representation_item	45		thread <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'hand'} descriptive_representation_item
defined_thread to feature_ parameter (as crest)	PATH			thread <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'crest'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
defined_thread to feature_ parameter (as density)	PATH			thread <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation => shape_representation_with_parameters} representation.items[i] ->> representation_item => {representation_item.name = 'density'} measure_representation_item {measure_representation_item <= measure_with_unit => ratio_measure_with_unit}
defined_thread to feature_parameter (as major_diameter)	PATH			thread <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] ->> representation_item => {representation_item.name = 'major diameter'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
defined_thread to feature_ parameter (as minor_ diameter)	PATH			thread <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'minor diameter'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
defined_thread to feature_ parameter (as pitch_ diameter)	PATH			thread <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'pitch diameter'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
DIAMETER_TAPER	taper	214		taper <= shape_aspect {[shape_aspect.description = 'diameter taper'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
diameter_taper to feature_ parameter (as final_ diameter)	PATH			taper <= shape_aspect {shape_aspect.description = 'diameter taper'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'final diameter'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
DIRECTED_TAPER	taper	214		taper <= shape_aspect {[shape_aspect.description = 'directed taper'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
draft_direction	direction_shape_ representation	214		taper <= shape_aspect {shape_aspect.description = 'directed taper'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => {representation.description = 'draft direction'} direction_shape_representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
directed_taper to feature_ parameter (as draft_angle)	PATH			taper <= shape_aspect {shape_aspect.description = 'directed taper'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'draft angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
EDGE_ROUND	edge_round	214		edge_round <= transition_feature <= shape_aspect
FACE_TERMINATION	joggle_termination	214		joggle_termination <= shape_aspect {[shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition] [shape_aspect.description = 'face termination']}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
FEATURE_ DEFINITION	feature_definition	214		feature_definition <= characterized_object
feature_ definition to cartesian_ coordinate_space_ 3d (as is_defined_ in)	PATH			feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.context_of_items -> representation_context => geometric_representation_context {geometric_representation_context.coordinate_space_dimension = '3'}
feature_ definition to axis_placement (as location_and_ orientation)	PATH			feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'orientation'} geometric_representation_item => placement
FEATURE_ OCCURRENCE	shape_aspect	41		
feature_function	shape_aspect. description	41		1)
feature_ occurrence to multi_language_ string (as feature_ function)	PATH		159	shape_aspect multi_language_attribute_item = shape_aspect multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
FILLET	fillet	214		fillet <= transition_feature <= shape_aspect
FLAT_BOTTOM	hole_bottom	214		hole_bottom <= shape_aspect {[shape_aspect.description = 'flat'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_object characterized_object => feature_component_definition]}
FLAT_SLOT_END_ TYPE	slot_end	214		slot_end <= shape_aspect {[shape_aspect.description = 'flat'] [shape_aspect.of_shape ->] product_definition_shape <= property_definition property_definition.definition ->] characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
flat_slot_end_ type to feature_ parameter (as first_ radius)	PATH			slot_end <= shape_aspect {shape_aspect.description = 'flat'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->] representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] ->] representation_item => {representation_item.name = 'first radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
flat_slot_end_ type to feature_ parameter (as second_ radius)	PATH			slot_end <= shape_aspect {shape_aspect.description = 'flat'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'second radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
FLAT_WITH_ CHAMFER_BOTTOM	hole_bottom	214		hole_bottom <= shape_aspect {[shape_aspect.description = 'flat with chamfer'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
flat_with_ chamfer_bottom to chamfer_ definition (as chamfer)	PATH			hole_bottom <= shape_aspect <- {shape_aspect.description = 'flat with chamfer'} shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship => {[(shape_aspect_relationship.name = 'first chamfer offset') (shape_aspect_relationship.name = 'second chamfer offset')] [shape_aspect_relationship.related_shape_aspect -> shape_aspect => chamfer_offset]} feature_component_relationship
FLAT_WITH_RADIUS_ BOTTOM	hole_bottom	214		hole_bottom <= shape_aspect {[shape_aspect.description = 'flat with radius'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
flat_with_radius_ bottom to feature_ parameter (as radius)	PATH			hole_bottom <= shape_aspect {shape_aspect.description = 'flat with radius'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation.items[i] -> representation_item => {representation_item.name = 'radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
FORM_FEATURE_IN_PANEL	feature_in_panel	214		feature_in_panel <= feature_definition <= characterized_object
feature_type	characterized_object.description	41		feature_in_panel <= feature_definition <= characterized_object characterized_object.description {(characterized_object.description = 'pocket') (characterized_object.description = 'boss') (characterized_object.description = 'stairstep')}
form_feature_in_panel to angle_specification (as base_boundary_defined_by)	PATH			feature_in_panel <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'base boundary'] [shape_aspect.description = 'directed angle occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'directed angle usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				directed_angle
form_feature_in_ panel to closed_ profile_usage (as base_ boundary_defined_ by)	PATH			feature_in_panel <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'base boundary'] [shape_aspect.description = 'closed profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship => {shape_aspect_relationship.name = 'closed profile usage'} shape_defining_relationship
form_feature_in_ panel to feature_ parameter (as base_fillet_ radius)	PATH			feature_in_panel <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'base fillet radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
form_feature_in_panel to angle_specification (as target_boundary_defined_by)	PATH			feature_in_panel <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'target boundary'] [shape_aspect.description = 'directed angle occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'directed angle usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => directed_angle
form_feature_in_panel to closed_profile_usage (as target_boundary_defined_by)	PATH			feature_in_panel <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <-

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'target boundary'] [shape_aspect.description = 'closed profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship => {shape_aspect_relationship.name = 'closed profile usage'} shape_defining_relationship
form_feature_in_ panel to shell (as target_face)	PATH			feature_in_panel <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {[representation.description = 'target face'] [representation => connected_face_shape_representation]} representation.items[i] -> representation_item => topological_representation_item => connected_face_set => (closed_shell) (open_shell)
form_feature_in_ panel to surface (as target_face)	PATH			feature_in_panel <= feature_definition <= characterized_object characterized_definition = characterized_object

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {[representation.description = 'target face'] [representation => face_shape_representation]} representation.items[i] -> representation_item => geometric_representation_item => face_surface face_surface.face_geometry -> surface
form_feature_in_ panel to feature_ parameter (as target_ fillet_radius)	PATH			feature_in_panel <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'target fillet radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
GENERAL_BOSS	boss	214		boss <= feature_definition <= characterized_object {characterized_object.description = 'general'}
general_boss to general_closed_ profile (as boundary)	PATH			boss <= feature_definition <= characterized_object {characterized_object.description = 'general'} characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'boundary'] [shape_aspect.description = 'closed profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'closed profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => closed_path_profile

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
GENERAL_BOTTOM_ CONDITION	pocket_bottom	214		pocket_bottom <= shape_aspect {[shape_aspect.description = 'non-planar'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
general_bottom_ condition to shell (as floor_face)	PATH			pocket_bottom <= shape_aspect {shape_aspect.description = 'non-planar'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {[representation.description = 'floor face'] [representation => connected_face_shape_representation]} representation.items[i] -> representation_item => topological_representation_item => connected_face_set => (closed_shell) (open_shell)
general_bottom_ condition to surface (as floor_face)	PATH			pocket_bottom <= shape_aspect {shape_aspect.description = 'non-planar'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_representation -> representation {[representation.description = 'floor face'] [representation => face_shape_representation]} representation.items[i] -> representation_item => geometric_representation_item => face_surface face_surface.face_geometry -> surface
general_bottom_ condition to feature_ parameter (as floor_ radius)	PATH			pocket_bottom <= shape_aspect {shape_aspect.description = 'non-planar'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'floor radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
GENERAL_CLOSED_ PROFILE	closed_path_profile	214		closed_path_profile <= shape_aspect {shape_aspect.of_shape -> product_definition_shape <=

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition}
general_closed_ profile to curve (as profile_ shape)	PATH			closed_path_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {[representation.description = 'profile shape'] [representation => path_shape_representation]} representation.items[i] -> representation_item => {representation_item.name = 'profile curve'} geometric_representation_item => curve
GENERAL_OPEN_ PROFILE	open_path_profile	214		open_path_profile <= shape_aspect {shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
general_open_ profile to curve (as profile_ shape)	PATH			open_path_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {[representation.description = 'profile shape'] [representation => path_shape_representation]} representation.items[i] -> representation_item => {representation_item.name = 'profile curve'} geometric_representation_item => curve
GENERAL_PATH	path_feature_ component	214		path_feature_component <= shape_aspect {[shape_aspect.description = 'complex'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
general_path to curve (as path_ geometry)	PATH			path_feature_component <= shape_aspect {shape_aspect.description = 'complex'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{[representation.description = 'path geometry'] [representation => path_shape_representation]} representation.items[i] -> representation_item => {representation_item.name = 'path curve'} geometric_representation_item => curve
GENERAL_PATTERN	feature_pattern	214		feature_pattern <= replicate_feature <= feature_definition <= characterized_object
general_pattern to axis_ placement (as added_ instance)	PATH			feature_pattern <= replicate_feature <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'added instance placement'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				geometric_representation_item => placement
GENERAL_POCKET_ IN_SOLID	pocket	214		pocket <= feature_definition <= characterized_object {characterized_object.description = 'non-rectangular'}
general_pocket_ in_solid to profile (as boundary)	PATH			pocket <= feature_definition <= characterized_object {characterized_object.description = 'non-rectangular'} characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'boundary'] [shape_aspect.description = 'profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => (circular_closed_profile) (closed_path_profile) (ngon_closed_profile) (rectangular_closed_profile) (open_path_profile) (partial_circular_profile) (rounded_u_profile) (square_u_profile) (tee_profile) (vee_profile)

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
GENERAL_TOP_ CONDITION	boss_top	214		boss_top <= shape_aspect {[shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition] [shape_aspect.description = 'non-planar']}
general_top_ condition to shell (as top_face)	PATH			boss_top <= shape_aspect {shape_aspect.description = 'non-planar' } represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {[representation.description = 'top face'] [representation => connected_face_shape_representation]} representation.items[i] -> representation_item => topological_representation_item => connected_face_set => (closed_shell) (open_shell)
general_top_ condition to surface (as top_face)	PATH			boss_top <= shape_aspect {shape_aspect.description = 'non-planar' } represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_representation -> representation {[representation.description = 'top face'] [representation => face_shape_representation]} representation.items[i] -> representation_item => geometric_representation_item => face_surface face_surface.face_geometry -> surface
HOLE_BOTTOM_ CONDITION	hole_bottom	214		hole_bottom <= shape_aspect {(shape_aspect.description = 'conical') (shape_aspect.description = 'flat') (shape_aspect.description = 'flat with radius') (shape_aspect.description = 'spherical') (shape_aspect.description = 'through') (shape_aspect.description = 'flat with chamfer')}
is_applied_to	descriptive_ representation_item	45		hole_bottom <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'bottom location'} descriptive_representation_item

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{(descriptive_representation_item.description = 'path start') (descriptive_representation_item.description = 'path end')}
HOLE_FEATURE_IN_ PANEL	hole_in_panel	214		hole_in_panel <= feature_definition <= characterized_object
hole_type	characterized_object. description	41		hole_in_panel <= feature_definition <= characterized_object characterized_object.description
hole_feature_in_ panel to closed_ profile_usage (as hole_ profile)	PATH			hole_in_panel <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'hole profile'] [shape_aspect.description = 'closed profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship => {shape_aspect_relationship.name = 'closed profile usage'} shape_defining_relationship
hole_feature_in_ panel to angle_ taper (as taper)	PATH			hole_in_panel <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'taper']

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				[shape_aspect.description = 'taper occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'taper usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'angle taper'} taper
hole_feature_in_ panel to diameter_taper (as taper)	PATH			hole_in_panel <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'taper'] [shape_aspect.description = 'taper occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'taper usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'diameter taper'} taper
hole_feature_in_ panel to directed_taper (as taper)	PATH			hole_in_panel <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <-

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'taper'] [shape_aspect.description = 'taper occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'taper usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'directed taper'} taper
JOGGLE_FEATURE	joggle	214		joggle <= feature_definition <= characterized_object
joggle_feature to linear_path (as defining_ path)	PATH			joggle <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'defining path'] [shape_aspect.description = 'path feature component occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'path feature component usage'] [shape_aspect_relationship => shape_defining_relationship]}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'linear'} path_feature_component
joggle_feature to feature_ parameter (as direction_ angle_at_path_ end)	PATH			joggle <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'direction angle at path end'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
joggle_feature to feature_ parameter (as direction_ angle_at_path_ start)	PATH			joggle <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'direction angle at path start'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
joggle_feature to feature_ parameter (as height)	PATH			joggle <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'height'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
joggle_feature to feature_ parameter (as slope_angle_ at_path_end)	PATH			joggle <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation => shape_representation_with_parameters} representation.items[i] ->> representation_item => {representation_item.name = 'slope angle at path end'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
joggle_feature to feature_ parameter (as slope_angle_ at_path_start)	PATH			joggle <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] ->> representation_item => {representation_item.name = 'slope angle at path start'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
joggle_feature to joggle_ termination (as termination)	PATH			feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'termination'] [shape_aspect.description = 'joggle termination occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'joggle termination usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {(shape_aspect.description = 'value termination') (shape_aspect.description = 'face termination')} joggle_termination
JOGGLE_ TERMINATION	joggle_termination	214		joggle_termination <= shape_aspect {[shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition] [(shape_aspect.description = 'value termination') (shape_aspect.description = 'face termination')]}
LINEAR_BEAD_END_ TYPE	bead_end	214		bead_end <= shape_aspect {[shape_aspect.description = 'linear']

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				[shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
end_length	measure_ representation_item	45		bead_end <= shape_aspect {shape_aspect.description = 'linear'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'end length'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
end_shape	descriptive_ representation_item	45		bead_end <= shape_aspect {shape_aspect.description = 'linear'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'end shape'} descriptive_representation_item {(descriptive_representation_item.description = 'straight') (descriptive_representation_item.description = 'tapered')}
LINEAR_PATH	path_feature_ component	214		path_feature_component <= shape_aspect {[shape_aspect.description = 'linear'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
linear_path to feature_ parameter (as distance)	PATH			path_feature_component <= shape_aspect {shape_aspect.description = 'linear'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'distance'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
linear_path to axis_placement (as is_defined_by)	PATH			path_feature_component <= shape_aspect {shape_aspect.description = 'linear'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'orientation'} geometric_representation_item => placement
LOCATOR_FEATURE	locator	214		locator <= feature_definition <= characterized_object
locator_feature to feature_parameter (as diameter)	PATH			locator <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <-

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'diameter'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
locator_feature to feature_ parameter (as tip_angle)	PATH			locator <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'tip angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
locator_feature to feature_ parameter (as tip_radius)	PATH			locator <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'tip radius'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
NGON_PROFILE	ngon_closed_profile	214		ngon_closed_profile <= shape_aspect {shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition}
number_of_sides	measure_ representation_item	45		ngon_closed_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'number of sides'} measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
ngon_profile to feature_ parameter (as circumscribed_ radius)	PATH			ngon_closed_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'circumscribed diameter'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
ngon_profile to feature_ parameter (as corner_ radius)	PATH			ngon_closed_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'corner radius'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
OPEN_BEAD_END_ TYPE	bead_end	214		bead_end <= shape_aspect {[shape_aspect.description = 'open'] [shape_aspect.of_shape ->] product_definition_shape <= property_definition property_definition.definition ->] characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
OPEN_PROFILE	(open_path_profile) (partial_circular_ profile) (rounded_u_profile) (square_u_profile) (tee_profile) (vee_profile)	214 214 214 214 214 214		(open_path_profile <=) (partial_circular_profile <=) (rounded_u_profile <=) (square_u_profile <=) (tee_profile <=) (vee_profile <=) shape_aspect {shape_aspect.of_shape ->] product_definition_shape <= property_definition property_definition.definition ->] characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition}
OPEN_SLOT_END_ TYPE	slot_end	214		slot_end <= shape_aspect {[shape_aspect.description = 'open'] [shape_aspect.of_shape ->] product_definition_shape <= property_definition

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
PANEL_HOLE_WITH_ BARRING	hole_in_panel	214		hole_in_panel <= feature_definition <= characterized_object
panel_hole_with_ barring to closed_profile_ usage (as barring_ profile)	PATH			hole_in_panel <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'barring profile'] [shape_aspect.description = 'closed profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship => {shape_aspect_relationship.name = 'closed profile usage'} shape_defining_relationship
PARTIAL_CIRCULAR_ PATH	path_feature_ component	214		path_feature_component <= shape_aspect {[shape_aspect.description = 'partial circular'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
partial_circular_ path to feature_ parameter (as sweep_angle)	PATH			path_feature_component <= shape_aspect {shape_aspect.description = 'partial circular'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'sweep angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
PARTIAL_CIRCULAR_ PROFILE	partial_circular_ profile	214		partial_circular_profile <= shape_aspect {shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
partial_circular_ profile to feature_ parameter (as radius)	PATH			partial_circular_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
partial_circular_ profile to feature_ parameter (as sweep_angle)	PATH			partial_circular_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'sweep angle'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
PATH	path_feature_ component	214		path_feature_component <= shape_aspect {shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition}
path to cartesian_ coordinate_space (as is_defined_ in)	PATH			path_feature_component <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.context_of_items -> representation_context => geometric_representation_context
PLANAR_BOTTOM_ CONDITION	pocket_bottom	214		pocket_bottom <= shape_aspect {[shape_aspect.description = 'planar'] [shape_aspect.of_shape -> product_definition_shape <= property_definition

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
floor_location	location_shape_representation	214		pocket_bottom <= shape_aspect {shape_aspect.description = 'planar'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => {representation.description = 'floor location'} location_shape_representation
floor_normal	direction_shape_representation	214		pocket_bottom <= shape_aspect {shape_aspect.description = 'planar'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => {representation.description = 'floor normal'} direction_shape_representation
planar_bottom_condition to feature_parameter (as floor_radius)	PATH			pocket_bottom <= shape_aspect {shape_aspect.description = 'planar'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'floor radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
PLANAR_TOP_ CONDITION	boss_top	214		boss_top <= shape_aspect {[shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition] [shape_aspect.description = 'planar']}
top_location	location_shape_ representation	214		boss_top <= shape_aspect {shape_aspect.description = 'planar' } represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => {representation.description = 'top location'} location_shape_representation
top_normal	direction_shape_ representation	214		boss_top <= shape_aspect

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{shape_aspect.description = 'planar' } represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => {representation.description = 'top normal'} direction_shape_representation
POCKET_BOTTOM_ CONDITION	pocket_bottom	214		pocket_bottom <= shape_aspect {[(shape_aspect.description = 'planar') (shape_aspect.description = 'non-planar') (shape_aspect.description = 'through')] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
is_applied_to	descriptive_ representation_item	45		pocket_bottom <= shape_aspect {(shape_aspect.description = 'planar') (shape_aspect.description = 'non-planar') (shape_aspect.description = 'through')} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation => shape_representation_with_parameters} representation.items[i] ->> representation_item => {representation_item.name = 'bottom location'} descriptive_representation_item {(descriptive_representation_item.description = 'path start') (descriptive_representation_item.description = 'path end')}
POCKET_FEATURE_ IN_SOLID	pocket	214		pocket <= feature_definition <= characterized_object {(characterized_object.description = 'open rectangular') (characterized_object.description = 'closed rectangular') (characterized_object.description = 'non-rectangular')}
pocket_feature_ in_solid to pocket_bottom_ condition (as bottom_ condition)	PATH			pocket <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'bottom condition'] [shape_aspect.description = 'pocket bottom occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'pocket bottom usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect ->> shape_aspect => {(shape_aspect.description = 'planar') (shape_aspect.description = 'non-planar') (shape_aspect.description = 'through')}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				pocket_bottom
pocket_feature_ in_solid to transformation (as profile_ placement)	PATH			pocket <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape [shape_aspect <- {[shape_aspect.name = 'boundary'] [shape_aspect.description = 'profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect {shape_aspect => (circular_closed_profile) (closed_path_profile) (ngon_closed_profile) (rectangular_closed_profile) (open_path_profile) (partial_circular_profile) (rounded_u_profile) (square_u_profile) (tee_profile) (vee_profile)} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item <- {[representation_item.name = 'orientation'] [representation_item => geometric_representation_item => placement]} representation_map.mapping_origin] [represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item <- {[representation_item.name = 'profile placement'] [representation_item => geometric_representation_item => placement]} mapped_item.mapping_target]

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
pocket_feature_ in_solid to linear_path (as sweep_path)	PATH			pocket <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'sweep path'] [shape_aspect.description = 'path feature component occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'path feature component usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'linear'} path_feature_component
pocket_feature_ in_solid to angle_taper (as taper)	PATH			pocket <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'taper'] [shape_aspect.description = 'taper occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'taper usage']

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				[shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'angle taper'} taper
pocket_feature_ in_solid to diameter_taper (as taper)	PATH			pocket <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'taper'] [shape_aspect.description = 'taper occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'taper usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'diameter taper'} taper
pocket_feature_ in_solid to directed_taper (as taper)	PATH			pocket <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect <- {[shape_aspect.name = 'taper'] [shape_aspect.description = 'taper occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'taper usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'directed taper'} taper
PROCESS_FEATURE_ IN_PANEL	(bead) (hole_in_panel) (feature_in_panel) (joggle) (locator) (compound_feature_in_ panel)	214 214 214 214 214 214		(bead <=) (hole_in_panel <=) (feature_in_panel <=) (joggle <=) (locator <=) (compound_feature_in_panel <=) feature_definition <= characterized_object
punch_direction	direction_shape_ representation	214		(bead <=) (hole_in_panel <=) (feature_in_panel <=) (joggle <=) (locator <=) (compound_feature_in_panel <=) feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <-

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->\ representation =>\ {representation.description = 'punch direction'} direction_shape_representation
PROCESS_FEATURE_IN_SOLID	(slot) (pocket) (boss) (round_hole) (rib) (compound_feature_in_solid)	214 214 214 214 214 214		(slot <=) (pocket <=) (boss <=) (round_hole <=) (rib <=) (compound_feature_in_solid <=) feature_definition <= characterized_object
PROFILE	(circular_closed_profile) (closed_path_profile) (ngon_closed_profile) (rectangular_closed_profile) (open_path_profile) (partial_circular_profile) (rounded_u_profile) (square_u_profile) (tee_profile) (vee_profile)	214 214 214 214 214 214 214 214 214 214		(circular_closed_profile <=) (closed_path_profile <=) (ngon_closed_profile <=) (rectangular_closed_profile <=) (open_path_profile <=) (partial_circular_profile <=) (rounded_u_profile <=) (square_u_profile <=) (tee_profile <=) (vee_profile <=) shape_aspect {shape_aspect.of_shape ->\ product_definition_shape <= property_definition property_definition.definition ->\ characterized_definition characterized_definition = characterized_object characterized_object =>\ feature_component_definition}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
profile to cartesian_ coordinate_space (as is_defined_ in)	PATH			(circular_closed_profile <=) (closed_path_profile <=) (ngon_closed_profile <=) (rectangular_closed_profile <=) (open_path_profile <=) (partial_circular_profile <=) (rounded_u_profile <=) (square_u_profile <=) (tee_profile <=) (vee_profile <=) shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.context_of_items -> representation_context => geometric_representation_context
profile to axis_ placement (as parameter_ reference)	PATH			(circular_closed_profile <=) (closed_path_profile <=) (ngon_closed_profile <=) (rectangular_closed_profile <=) (open_path_profile <=) (partial_circular_profile <=) (rounded_u_profile <=) (square_u_profile <=) (tee_profile <=) (vee_profile <=) shape_aspect

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'orientation'} geometric_representation_item => placement
RADIUSED_BEAD_ END_TYPE	bead_end	214		bead_end <= shape_aspect {[shape_aspect.description = 'radiused'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
RADIUSED_SLOT_ END_TYPE	slot_end	214		slot_end <= shape_aspect {[shape_aspect.description = 'radiused'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_object => feature_component_definition]}
RECTANGULAR_BOSS	boss	214		boss <= feature_definition <= characterized_object {characterized_object.description = 'rectangular'}
rectangular_boss to rectangular_ closed_profile (as boundary)	PATH			boss <= feature_definition <= characterized_object {characterized_object.description = 'rectangular'} characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'boundary'] [shape_aspect.description = 'closed profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'closed profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => rectangular_closed_profile
RECTANGULAR_ CLOSED_POCKET_IN_ SOLID	pocket	214		pocket <= feature_definition <= characterized_object {characterized_object.description = 'closed rectangular'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_ closed_pocket_in_ solid to rectangular_ closed_profile (as boundary)	PATH			pocket <= feature_definition <= characterized_object {characterized_object.description = 'closed rectangular'} characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'boundary'] [shape_aspect.description = 'closed profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'closed profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => rectangular_closed_profile
RECTANGULAR_ CLOSED_PROFILE	rectangular_closed_ profile	214		rectangular_closed_profile <= shape_aspect {shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
rectangular_ closed_profile to feature_ parameter (as corner_ radius)	PATH			rectangular_closed_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'corner radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
rectangular_ closed_profile to feature_ parameter (as profile_ length)	PATH			rectangular_closed_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'length'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
rectangular_ closed_profile to feature_ parameter (as profile_ width)	PATH			rectangular_closed_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'width'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
RECTANGULAR_ OFFSET_PATTERN	shape_aspect	41		{shape_aspect [shape_aspect.description = 'rectangular offset'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object <- {characterized_object => feature_definition => replicate_feature =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				modified_pattern} characterized_object_relationship.related_object characterized_object_relationship {characterized_object_relationship.name = 'replicate base assignment'} characterized_object_relationship.relating_object->characterized_object => feature_definition => replicate_feature => rectangular_pattern]}
column_index	measure_representation_item	45		shape_aspect {shape_aspect.description = 'rectangular offset'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'column index'} measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure}
offset_direction	direction_shape_representation	214		shape_aspect {shape_aspect.description = 'rectangular offset'} represented_definition = shape_aspect represented_definition <-

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->
representation => {representation.description = 'offset direction'} direction_shape_representation
row_index	measure_ representation_item	45		shape_aspect {shape_aspect.description = 'rectangular offset'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->
representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] ->
representation_item => {representation_item.name = 'row index'} measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component ->
measure_value measure_value = count_measure count_measure}
rectangular_ offset_pattern to feature_ parameter (as offset_ distance)	PATH			shape_aspect {shape_aspect.description = 'rectangular offset'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'offset distance'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
RECTANGULAR_OMIT_ PATTERN	shape_aspect	41		{shape_aspect [shape_aspect.description = 'rectangular omit'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object <- {characterized_object => feature_definition => replicate_feature => modified_pattern} characterized_object_relationship.related_object characterized_object_relationship {characterized_object_relationship.name = 'replicate base assignment'} characterized_object_relationship.relating_object-> characterized_object => feature_definition => replicate_feature => rectangular_pattern]}
column_index	measure_ representation_item	45		shape_aspect {shape_aspect.description = 'rectangular omit'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'column index'} measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure}
row_index	measure_ representation_item	45		shape_aspect {shape_aspect.description = 'rectangular omit'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{representation_item.name = 'row index'} measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure}
RECTANGULAR_OPEN_ POCKET_IN_SOLID	pocket	214		pocket <= feature_definition <= characterized_object {characterized_object.description = 'open rectangular'}
rectangular_open_ pocket_in_solid to square_u_ profile (as boundary)	PATH			pocket <= feature_definition <= characterized_object {characterized_object.description = 'open rectangular'} characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'boundary'] [shape_aspect.description = 'open profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'open profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => square_u_profile
RECTANGULAR_ PATTERN	rectangular_pattern	214		rectangular_pattern <= replicate_feature <= feature_definition <=

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_object
column_layout_direction	direction_shape_representation	214		rectangular_pattern <= replicate_feature <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => {representation.description = 'column layout direction'} direction_shape_representation
columns	measure_representation_item	45		rectangular_pattern <= replicate_feature <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'number of columns'} measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure}
row_layout_direction	direction_shape_representation	214		rectangular_pattern <= replicate_feature <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => {representation.description = 'row layout direction'} direction_shape_representation
rows	measure_representation_item	45		rectangular_pattern <= replicate_feature <= feature_definition <= characterized_object

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'number of rows'} measure_representation_item {measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure count_measure}
rectangular_ pattern to feature_ parameter (as column_ spacing)	PATH			rectangular_pattern <= replicate_feature <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'column spacing'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
rectangular_ pattern to rectangular_omit_ pattern (as missing_ feature)	PATH			rectangular_pattern <= replicate_feature <= feature_definition <= characterized_object <- characterized_object_relationship.relating_object characterized_object_relationship {characterized_object_relationship.name = 'replicate base assignment'} characterized_object_relationship.related_object -> characterized_object {characterized_object => feature_definition => replicate_feature => modified_pattern} characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_shape <- shape_aspect.of_shape shape_aspect {shape_aspect.description = 'rectangular omit'}
rectangular_ pattern to rectangular_ offset_pattern (as relocated_ feature)	PATH			rectangular_pattern <= replicate_feature <= feature_definition <= characterized_object <- characterized_object_relationship.relating_object characterized_object_relationship {characterized_object_relationship.name = 'replicate base assignment'} characterized_object_relationship.related_object -> characterized_object {characterized_object => feature_definition => replicate_feature => modified_pattern} characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect {shape_aspect.description = 'rectangular offset'}
rectangular_ pattern to feature_ parameter (as row_spacing)	PATH			rectangular_pattern <= replicate_feature <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'row spacing'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
REPLICATE_BASE_ ASSIGNMENT	characterized_object_ relationship	41		{characterized_object_relationship characterized_object_relationship.name = 'replicate base assignment'}
replicate_base_ assignment to feature_ definition (as base_ feature)	PATH			characterized_object_relationship {characterized_object_relationship.name = 'replicate base assignment'} characterized_object_relationship.relating_object -> characterized_object => feature_definition
replicate_base_ assignment to transformation (as transformation)	PATH			characterized_object_relationship {characterized_object_relationship.name = 'replicate base assignment'} [characterized_object_relationship.relating_object -> characterized_object {characterized_object => feature_definition } characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item <- {[representation_item.name = 'orientation'] [representation_item => geometric_representation_item => placement]} representation_map.mapping_origin] [characterized_object_relationship.related_object -> characterized_object {characterized_object => feature_definition => replicate_feature} characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item <- {[representation_item.name = 'replicate base placement'] [representation_item => geometric_representation_item => placement]} mapped_item.mapping_target]
REPLICATE_ FEATURE	replicate_feature	214		replicate_feature <= feature_definition <= characterized_object
replicate_ feature to replicate_base_ assignment (as replicate_ base)	PATH			replicate_feature <= feature_definition <= characterized_object <- characterized_object_relationship.related_object characterized_object_relationship {characterized_object_relationship.name = 'replicate base assignment'}
RIB_FEATURE	rib	214		rib <= feature_definition <= characterized_object
rib_feature to directed_taper (as draft_taper)	PATH			rib <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'taper']

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				[shape_aspect.description = 'taper occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'taper usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'directed taper'} taper
rib_feature to feature_ parameter (as fillet_ radius)	PATH			rib <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'fillet radius'} measure_representation_item {measure_representation_item <= measure_with_unit =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				length_measure_with_unit}
rib_feature to transformation (as profile_ placement)	PATH			rib <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape [shape_aspect <- {[shape_aspect.name = 'rib profile'] [shape_aspect.description = 'open profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'open profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect {shape_aspect => (open_path_profile) (partial_circular_profile) (rounded_u_profile) (square_u_profile) (tee_profile) (vee_profile)} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item <- {[representation_item.name = 'orientation'] [representation_item => geometric_representation_item => placement]} representation_map.mapping_origin] [represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item <- {[representation_item.name = 'profile placement'] [representation_item => geometric_representation_item => placement]} mapped_item.mapping_target]
rib_feature to open_profile (as rib_profile)	PATH			rib <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <-

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'rib profile'] [shape_aspect.description = 'open profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'open profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => (open_path_profile) (partial_circular_profile) (rounded_u_profile) (square_u_profile) (tee_profile) (vee_profile)
rib_feature to feature_ parameter (as shoulder_ radius)	PATH			rib <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'shoulder radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
rib_feature to path (as sweep_path)	PATH			rib <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'sweep path'] [shape_aspect.description = 'path feature component occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'path feature component usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => path_feature_component
ROUND_HOLE	round_hole	214		round_hole <= feature_definition <= characterized_object

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
round_hole to hole_bottom_condition (as bottom_condition)	PATH			round_hole <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'bottom condition'] [shape_aspect.description = 'hole bottom occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'hole bottom usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {(shape_aspect.description = 'conical') (shape_aspect.description = 'flat') (shape_aspect.description = 'flat with radius') (shape_aspect.description = 'spherical') (shape_aspect.description = 'through') (shape_aspect.description = 'flat with chamfer')} hole_bottom
round_hole to circular_closed_profile (as boundary)	PATH			round_hole <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <-

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{[shape_aspect.name = 'boundary'] [shape_aspect.description = 'closed profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'closed profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => circular_closed_profile
round_hole to transformation (as profile_ placement)	PATH			round_hole <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape [shape_aspect <- {[shape_aspect.name = 'boundary'] [shape_aspect.description = 'closed profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'closed profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect {shape_aspect => circular_closed_profile} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item <- {[representation_item.name = 'orientation'] [representation_item => geometric_representation_item => placement]} representation_map.mapping_origin] [represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item <- {[representation_item.name = 'profile placement'] [representation_item => geometric_representation_item => placement]} mapped_item.mapping_target]
round_hole to linear_path (as sweep_path)	PATH			round_hole <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <-

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'sweep path'] [shape_aspect.description = 'path feature component occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'path feature component usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'linear'} path_feature_component
round_hole to angle_taper (as taper)	PATH			round_hole <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'taper'] [shape_aspect.description = 'taper occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'taper usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'angle taper'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				taper
round_hole to diameter_taper (as taper)	PATH			round_hole <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'taper'] [shape_aspect.description = 'taper occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'taper usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'diameter taper'} taper
round_hole to directed_taper (as taper)	PATH			round_hole <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'taper'] [shape_aspect.description = 'taper occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{[shape_aspect_relationship.name = 'taper usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.description = 'directed taper'} taper
round_hole to chamfer_ definition (as top_chamfer)	PATH			round_hole <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {shape_aspect.description = 'top chamfer'} shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship => {[(shape_aspect_relationship.name = 'first chamfer offset') (shape_aspect_relationship.name = 'second chamfer offset')] [shape_aspect_relationship.related_shape_aspect -> shape_aspect => chamfer_offset]} feature_component_relationship
ROUNDED_U_ PROFILE	rounded_u_profile	214		rounded_u_profile <= shape_aspect {shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
rounded_u_ profile to feature_ parameter (as width)	PATH			rounded_u_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'width'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
SLOT_END_TYPE	slot_end	214		slot_end <= shape_aspect {[(shape_aspect.description = 'flat') (shape_aspect.description = 'open') (shape_aspect.description = 'radiused') (shape_aspect.description = 'woodruff')] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
is_applied_to	shape_aspect.name	41		slot_end <= shape_aspect <-

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{(shape_aspect.description = 'flat') (shape_aspect.description = 'open') (shape_aspect.description = 'radiused') (shape_aspect.description = 'woodruff')} shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship {shape_aspect_relationship.name = 'slot end usage'} shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect.description = 'slot end occurrence'} shape_aspect.name {(shape_aspect.name = 'end condition at path start') (shape_aspect.name = 'end condition at path end')}
SLOT_FEATURE	slot	214		slot <= feature_definition <= characterized_object
maximum_depth	measure_ representation_item	45		slot <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation.items[i] -> representation_item => {representation_item.name = 'depth'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
slot_feature to slot_end_type (as end_ condition)	PATH			slot <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[(shape_aspect.name = 'end condition at path start') (shape_aspect.name = 'end condition at path end')] [shape_aspect.description = 'slot end occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'slot end usage'] [shape_aspect_relationship => feature_component_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {(shape_aspect.description = 'flat') (shape_aspect.description = 'open') (shape_aspect.description = 'radiused') (shape_aspect.description = 'woodruff')} slot_end

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
slot_feature to transformation (as profile_ placement)	PATH			slot <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape [shape_aspect <- {[shape_aspect.name = 'sweep shape'] [shape_aspect.description = 'open profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'open profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect {shape_aspect => (open_path_profile) (partial_circular_profile) (rounded_u_profile) (square_u_profile) (tee_profile) (vee_profile)} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation => shape_representation_with_parameters} representation.items[i] ->> representation_item <- {[representation_item.name = 'orientation'] [representation_item => geometric_representation_item => placement]} representation_map.mapping_origin] [represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->> representation {representation => shape_representation} representation.items[i] ->> representation_item <- {[representation_item.name = 'profile placement'] [representation_item => geometric_representation_item => placement]} mapped_item.mapping_target]
slot_feature to open_profile (as slot_profile)	PATH			slot <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect <- {[shape_aspect.name = 'sweep shape'] [shape_aspect.description = 'open profile occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'open profile usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => (open_path_profile) (partial_circular_profile) (rounded_u_profile) (square_u_profile) (tee_profile) (vee_profile)
slot_feature to path (as sweep_path)	PATH			slot <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- {[shape_aspect.name = 'sweep path'] [shape_aspect.description = 'path feature component occurrence']} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'path feature component usage'] [shape_aspect_relationship => shape_defining_relationship]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => path_feature_component

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
SPHERICAL_BOTTOM	hole_bottom	214		hole_bottom <= shape_aspect {[shape_aspect.description = 'spherical'] [shape_aspect.of_shape ->] product_definition_shape <= property_definition property_definition.definition ->] characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
spherical_bottom to feature_ parameter (as radius)	PATH			hole_bottom <= shape_aspect {shape_aspect.description = 'spherical'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->] representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] ->] representation_item => {representation_item.name = 'radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
SQUARE_U_PROFILE	square_u_profile	214		square_u_profile <= shape_aspect {shape_aspect.of_shape ->] product_definition_shape <=

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition}
square_u_profile to feature_ parameter (as first_angle)	PATH			square_u_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'first angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
square_u_profile to feature_ parameter (as first_ radius)	PATH			square_u_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{representation => shape_representation => shape_representation_with_parameters} representation.items[i] ->> representation_item => {representation_item.name = 'first radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
square_u_profile to feature_parameter (as second_angle)	PATH			square_u_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] ->> representation_item => {representation_item.name = 'second angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
square_u_profile to feature_parameter (as second_radius)	PATH			square_u_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] ->representation_item => {representation_item.name = 'second radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
square_u_profile to feature_parameter (as width)	PATH			square_u_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] ->representation_item => {representation_item.name = 'width'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
TEE_PROFILE	tee_profile	214		tee_profile <= shape_aspect

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition}
tee_profile to feature_ parameter (as cross_bar_ depth)	PATH			tee_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'cross bar depth'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
tee_profile to feature_ parameter (as cross_bar_ width)	PATH			tee_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'cross bar width'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
tee_profile to feature_ parameter (as depth)	PATH			tee_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'depth'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
tee_profile to feature_ parameter (as first_angle)	PATH			tee_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'first angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
tee_profile to feature_ parameter (as first_ offset)	PATH			tee_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'first offset'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
tee_profile to feature_ parameter (as radius)	PATH			tee_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
tee_profile to feature_ parameter (as second_ angle)	PATH			tee_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'second angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
tee_profile to feature_ parameter (as second_ offset)	PATH			tee_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'second offset'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
tee_profile to feature_ parameter (as width)	PATH			tee_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'width'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
THREAD_FEATURE	thread_occurrence	214		thread_occurrence <= shape_aspect
effective_thread_ length	measure_ representation_item	45		thread_occurrence <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'effective length'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
inner	shape_aspect. description	41		thread_occurrence <= shape_aspect

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect.description {(shape_aspect.description = 'inner') (shape_aspect.description = 'outer')}
maximum_thread_ length	measure_ representation_item	45		thread_occurrence <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'maximum length'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
thread_feature to axis_ placement (as orientation)	PATH			thread_occurrence <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation.items[i] -> representation_item => {representation_item.name = 'orientation'} geometric_representation_item => placement
thread_feature to thread_ specification (as thread_ definition)	PATH			thread_occurrence <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {shape_aspect_relationship.name = 'thread usage'} shape_aspect_relationship.relating_shape_aspect -> shape_aspect shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_definition => (thread) (externally_defined_thread)
THREAD_ SPECIFICATION	(thread) (externally_defined_ thread)	214 214		(thread <= feature_definition <= characterized_object) (externally_defined_thread <= [feature_definition <= characterized_object] [externally_defined_item])
qualifier	descriptive_ representation_item	45		(thread <=) (externally_defined_thread <=) feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'qualifier'} descriptive_representation_item
THROUGH_BOTTOM_ CONDITION	pocket_bottom	214		pocket_bottom <= shape_aspect {[shape_aspect.description = 'through'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
THROUGH_HOLE_ CONDITION	hole_bottom	214		hole_bottom <= shape_aspect {[shape_aspect.description = 'through'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition ->

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}
TRANSITION_ FEATURE	transition_feature	214		transition_feature <= shape_aspect
transition_ feature to face (as face_1)	PATH			transition_feature <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->> representation {[representation.description = 'first face shape'] [representation => face_shape_representation]} representation.items[i] ->> representation_item => topological_representation_item => face => face_surface
transition_ feature to face (as face_2)	PATH			transition_feature <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->> representation {[representation.description = 'second face shape'] [representation => face_shape_representation]} representation.items[i] ->> representation_item => topological_representation_item =>

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				face => face_surface
VALUE_ TERMINATION	joggle_termination	214		joggle_termination <= shape_aspect {[shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition] [shape_aspect.description = 'value termination']}
value_ termination to feature_ parameter (as angle)	PATH			joggle_termination <= shape_aspect {shape_aspect.description = 'value termination'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
value_ termination to feature_ parameter (as depth)	PATH			joggle_termination <= shape_aspect {shape_aspect.description = 'value termination'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'depth'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
VEE_PROFILE	vee_profile	214		vee_profile <= shape_aspect {shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
vee_profile to feature_ parameter (as profile_ angle)	PATH			vee_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'profile angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
vee_profile to feature_ parameter (as radius)	PATH			vee_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'radius'}

Table 7 – Mapping table included_feature UoF (FF2) (continued)

Application element	AIM element	Source	Rules	Reference path
				measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
vee_profile to feature_ parameter (as tilt_angle)	PATH			vee_profile <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'tilt angle'} measure_representation_item {measure_representation_item <= measure_with_unit => plane_angle_measure_with_unit}
WOODRUFF_SLOT_ END_TYPE	slot_end	214		slot_end <= shape_aspect {[shape_aspect.description = 'woodruff'] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_component_definition]}

Table 7 – Mapping table included_feature UoF (FF2) (concluded)

Application element	AIM element	Source	Rules	Reference path
woodruff_slot_ end_type to feature_ parameter (as radius)	PATH			slot_end <= shape_aspect {shape_aspect.description = 'woodruff'} represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation => shape_representation_with_parameters} representation.items[i] -> representation_item => {representation_item.name = 'radius'} measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}

NOTE – The following note numbers correspond to the footnotemarks listed in the table above:

1 In case the attributes feature_occurrence.feature_function and shape_aspect.description are both instantiated in the ARM, they have to be combined into one attribute shape_aspect.description in the AIM.

Table 8 – Mapping table generative_featured_shape UoF (FF3)

Application element	AIM element	Source	Rules	Reference path
FEATURE_ OCCURRENCE	shape_aspect	41		{(shape_aspect group_member_item = shape_aspect group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment group_assignment.role -> group_role group_role.name = 'feature categorization') (shape_aspect => thread_occurrence) (shape_aspect => transition_feature) (shape_aspect => instanced_feature) (shape_aspect => placed_feature)}
feature_function	shape_aspect. description	41		1)
feature_ occurrence to multi_language_ string (as feature_ function)	PATH		159	shape_aspect multi_language_attribute_item = shape_aspect multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
FEATURE_ PLACEMENT	mapped_item	43		
feature_ placement to geometric_model (as placement_ reference)	PATH			mapped_item <= representation_item <- representation.items[i] representation <- {[representation.description = 'feature definition placement'] [representation => shape_representation]}

Table 8 – Mapping table generative_featured_shape UoF (FF3) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_representation property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = shape_aspect shape_aspect {shape_aspect => placed_feature} shape_aspect.of_shape -> product_definition_shape <= property_definition {property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation => shape_representation
feature_ placement to placed_feature (as placement_ reference)	PATH			(mapped_item <= representation_item <- representation.items[i] representation <- {[representation.description = 'feature definition placement'] [representation => shape_representation]} property_definition_representation.used_representation property_definition_representation

Table 8 – Mapping table generative_featured_shape UoF (FF3) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = shape_aspect shape_aspect <- {[shape_aspect => placed_feature] [shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition]} shape_aspect_relationship.related_shape_aspect shape_aspect_relationship {[shape_aspect_relationship.name = 'placement dependency'] [shape_aspect_relationship => dimensional_location => directed_dimensional_location]} shape_aspect_relationship.relating_shape_aspect -> shape_aspect => {shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition} placed_feature) (mapped_item mapped_item.mapping_target ->

Table 8 – Mapping table generative_featured_shape UoF (FF3) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item <- {representation_item => geometric_representation_item => placement} item_defined_transformation.transform_item_2 item_defined_transformation {item_defined_transformation.name = 'placement dependency'} item_defined_transformation.transform_item_1 -> representation_item <- {representation_item => geometric_representation_item => placement} mapped_item.mapping_target mapped_item <= representation_item representation.items[i] representation <- {[representation.description = 'feature definition placement'] [representation => shape_representation]} property_definition_representation.used_representation property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = shape_aspect shape_aspect => {shape_aspect.of_shape -> product_definition_shape <= property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition}

Table 8 – Mapping table generative_featured_shape UoF (FF3) (continued)

Application element	AIM element	Source	Rules	Reference path
				placed_feature)
feature_ placement to transformation (as transformation)	PATH			mapped_item [mapped_item.mapping_source -> representation_map {representation_map.mapped_representation -> representation <- {representation => shape_representation => shape_representation_with_parameters} property_definition_representation.used_representation property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => feature_definition} representation_map.mapping_origin {representation_map.mapping_origin -> representation_item => {representation_item.name = 'orientation'} geometric_representation_item => placement}] [mapped_item.mapping_target {mapped_item.mapping_target -> representation_item => geometric_representation_item => placement}]

Table 8 – Mapping table generative_featured_shape UoF (FF3) (continued)

Application element	AIM element	Source	Rules	Reference path
FEATURED_SHAPE	(featured_shape_representation) (product_definition_shape)	214 41		(featured_shape_representation <= shape_representation) ({product_definition_shape <= property_definition <- {property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition} property_definition_relationship.related_property_definition property_definition_relationship property_definition_relationship.name = 'featured shape sequence'})
featured_shape to item_shape (as base_shape)	PATH			(featured_shape_representation <= shape_representation <= representation <- representation_relationship.rep_2 representation_relationship {representation_relationship.name = 'feature application sequence'} representation_relationship.rep_1 -> representation <- {representation => shape_representation} property_definition_representation.used_representation property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = property_definition) (product_definition_shape <= property_definition <- {property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition

Table 8 – Mapping table generative_featured_shape UoF (FF3) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition} property_definition_relationship.related_property_definition property_definition_relationship {property_definition_relationship.name = 'featured shape sequence'} property_definition_relationship.relating_property_definition ->) property_definition => {property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition} product_definition_shape
PLACED_FEATURE	placed_feature	214		placed_feature <= shape_aspect {[represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {[representation.description = 'feature definition placement'] [representation => shape_representation]} representation.context_of_items ->] [shape_aspect.of_shape -> product_definition_shape <= property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation =>

Table 8 – Mapping table generative_featured_shape UoF (FF3) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.context_of_items ->] representation_context => geometric_representation_context}[2)]
placed_feature to feature_ definition (as definition)	PATH			placed_feature <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation <- {representation => shape_representation} property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition property_definition.definition -> characterized_definition

Table 8 – Mapping table generative_featured_shape UoF (FF3) (concluded)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = characterized_object characterized_object => feature_definition
placed_feature to feature_ placement (as is_placed_ by)	PATH			placed_feature <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {[representation.description = 'feature definition placement'] [representation => shape_representation]} representation.items[i] -> representation_item => mapped_item

NOTE – The following note numbers correspond to the footnotemarks listed in the table above:

1 In case the attributes feature_occurrence.feature_function and shape_aspect.description are both instantiated in the ARM, they have to be combined into one attribute shape_aspect.description in the AIM.

2 Sharing of representation_context.

Table 9 – Mapping table wireframe_model_2d UoF (G1)

Application element	AIM element	Source	Rules	Reference path
B_SPLINE_CURVE	b_spline_curve	503		
CARTESIAN_POINT	cartesian_point	503		
CIRCLE	circle	503		
COMPOSITE_CURVE	composite_curve	503		
CONSTRUCTIVE_ GEOMETRIC_ ELEMENT	(axis2_placement_2d) (cartesian_point) (point_on_curve) (curve)	503 503 503 503		
description	representation_item. name	503		(axis2_placement_2d <=) ((cartesian_point <=) (point_on_curve <=) point <=) (curve <=) geometric_representation_item <= representation_item representation_item.name
constructive_ geometric_ element to axis_ placement (as element)	PATH			axis2_placement_2d <= placement
constructive_ geometric_ element to curve (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to point (as element)	IDENTICAL MAPPING			
CONSTRUCTIVE_ GEOMETRY	constructive_ geometry_ representation	214		constructive_geometry_representation <= representation {representation.context_of_items -> representation_context => geometric_representation_context geometric_representation_context.coordinate_space_dimension = '2'}

Table 9 – Mapping table wireframe_model_2d UoF (G1) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometry to constructive_ geometric_ element (as element)	PATH			constructive_geometry_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => (axis2_placement_2d) (point => (cartesian_point) (point_on_curve)) (curve)
CONSTRUCTIVE_ GEOMETRY_ ASSOCIATION	constructive_ geometry_ representation_ relationship	214		constructive_geometry_representation_relationship <= representation_relationship
description	representation_ relationship. description	43		constructive_geometry_representation_relationship <= representation_relationship representation_relationship.description
constructive_ geometry_ association to constructive_ geometry (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => constructive_geometry_representation
constructive_ geometry_ association to geometric_model (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => shape_representation

Table 9 – Mapping table wireframe_model_2d UoF (G1) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometry_ association to constructive_ geometry (as constructive_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_2 -> constructive_geometry_representation
constructive_ geometry_ association to multi_language_ string (as description)	PATH		98	constructive_geometry_representation_relationship multi_language_attribute_item = \ constructive_geometry_representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation_relationship.description'}
CURVE	curve	503		
CURVE_SET_2D	geometric_curve_set	503		
curve_set_2d to curve (as element)	PATH			geometric_curve_set <= geometric_set geometric_set.elements -> geometric_set_select = curve
curve_set_2d to point (as element)	PATH			geometric_curve_set <= geometric_set geometric_set.elements -> geometric_set_select = point point => (cartesian_point) (point_on_curve)
DETAILED_ELEMENT	representation_item	503		{representation_item => (geometric_representation_item) (mapped_item)}
name	representation_item. name	503		

Table 9 – Mapping table wireframe_model_2d UoF (G1) (continued)

Application element	AIM element	Source	Rules	Reference path
detailed_element to multi_ language_string (as name)	PATH		154	representation_item multi_language_attribute_item = representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
DETAILED_ GEOMETRIC_MODEL_ ELEMENT	geometric_ representation_item	503		
DETAILED_MODEL_ ELEMENT	representation_item	503		{representation_item => (geometric_representation_item) (mapped_item)}
ELLIPSE	ellipse	503		
GEOMETRIC_MODEL	shape_representation	503	43,187	
description	representation. description	503		shape_representation <= representation representation.description
model_extent	uncertainty_measure_ with_unit	43		shape_representation <= representation => uncertainty_assigned_representation uncertainty_assigned_representation.uncertainty[i] -> uncertainty_measure_with_unit {uncertainty_measure_with_unit.name = 'model extent'} {uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit}
model_id	representation.id	503		shape_representation <= representation representation.id
role	representation.name	503		shape_representation <= representation representation.name {(representation.name) (representation.name = 'design shape')

Table 9 – Mapping table wireframe_model_2d UoF (G1) (continued)

Application element	AIM element	Source	Rules	Reference path
				(representation.name = 'idealized shape')}
geometric_model to template_ instance (as additional_ elements)	PATH			shape_representation <= representation representation.items[i] -> mapped_item
geometric_model to multi_ language_string (as description)	PATH		162	shape_representation multi_language_attribute_item = shape_representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation.description'}
geometric_model to cartesian_ coordinate_space (as is_defined_ in)	PATH			shape_representation => representation representation.context_of_items -> representation_context => geometric_representation_context {geometric_representation_context.coordinate_space_dimension = '2'}
GEOMETRIC_MODEL_ RELATIONSHIP	representation_ relationship	43		
description	representation_ relationship. description	43		
relation_type	representation_ relationship.name	43		
geometric_model_ relationship to multi_language_ string (as description)	PATH		155	representation_relationship multi_language_attribute_item = representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 9 – Mapping table wireframe_model_2d UoF (G1) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_model_ relationship to external_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => externally_defined_representation
geometric_model_ relationship to geometric_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => shape_representation
geometric_model_ relationship to external_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => externally_defined_representation
geometric_model_ relationship to geometric_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => shape_representation
GEOMETRIC_MODEL_ VERSION	shape_representation	503	181,33	{shape_representation version_item = shape_representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
version_id	identification_ assignment.assigned_ id	41	181,33	shape_representation version_item = shape_representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id

Table 9 – Mapping table wireframe_model_2d UoF (G1) (concluded)

Application element	AIM element	Source	Rules	Reference path
HYPERBOLA	hyperbola	503		
LINE	line	503		
OFFSET_CURVE	offset_curve_2d	503		
PARABOLA	parabola	503		
POINT	(cartesian_point) (point_on_curve)	503 503		
POINT_ON_CURVE	point_on_curve	503		
POLYLINE	polyline	503		
TRIMMED_CURVE	trimmed_curve	503		
WIREFRAME_MODEL_ 2D	geometrically_ bounded_2d_wireframe_ representation	503		geometrically_bounded_2d_wireframe_representation <= shape_representation
wireframe_model_ 2d to curve_set_ 2d (as element)	PATH			geometrically_bounded_2d_wireframe_representation <= shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => geometric_set => geometric_curve_set

Table 10 – Mapping table wireframe_model_3d UoF (G2)

Application element	AIM element	Source	Rules	Reference path
B_SPLINE_CURVE	b_spline_curve	510		
CARTESIAN_POINT	cartesian_point	510		
CIRCLE	circle	510		
COMPOSITE_CURVE	composite_curve	510		
CONSTRUCTIVE_ GEOMETRIC_ ELEMENT	(axis2_placement_3d) (cartesian_point) (point_on_curve) (point_replica) (curve)	510 510 510 510 510		
description	representation_item. name	510		(axis2_placement_3d <=) ((cartesian_point <=) (point_on_curve <=) (point_replica <=) point <=) (curve <=) geometric_representation_item <= representation_item representation_item.name
constructive_ geometric_ element to axis_ placement (as element)	PATH			axis2_placement_3d <= placement
constructive_ geometric_ element to curve (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to point (as element)	IDENTICAL MAPPING			
CONSTRUCTIVE_ GEOMETRY	constructive_ geometry_ representation	214		constructive_geometry_representation <= representation {representation.context_of_items -> representation_context => geometric_representation_context

Table 10 – Mapping table wireframe_model_3d UoF (G2) (continued)

Application element	AIM element	Source	Rules	Reference path
				geometric_representation_context.coordinate_space_dimension = '3'}
constructive_ geometry to constructive_ geometric_ element (as element)	PATH			constructive_geometry_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => (axis2_placement_3d) (point => (cartesian_point) (point_on_curve) (point_replica)) (curve)
CONSTRUCTIVE_ GEOMETRY_ ASSOCIATION	constructive_ geometry_ representation_ relationship	214		constructive_geometry_representation_relationship <= representation_relationship
description	representation_ relationship. description	43		constructive_geometry_representation_relationship <= representation_relationship representation_relationship.description
constructive_ geometry_ association to constructive_ geometry (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => constructive_geometry_representation
constructive_ geometry_ association to geometric_model (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => shape_representation

Table 10 – Mapping table wireframe_model_3d UoF (G2) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometry_ association to constructive_ geometry (as constructive_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_2 -> constructive_geometry_representation
constructive_ geometry_ association to multi_language_ string (as description)	PATH		98	constructive_geometry_representation_relationship multi_language_attribute_item = \ constructive_geometry_representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation_relationship.description'}
CURVE	curve	510		
CURVE_SET_3D	geometric_curve_set	510		
curve_set_3d to curve (as element)	PATH			geometric_curve_set <= geometric_set geometric_set.elements -> geometric_set_select = curve
curve_set_3d to point (as element)	PATH			geometric_curve_set <= geometric_set geometric_set.elements -> geometric_set_select = point point => (cartesian_point) (point_on_curve) (point_replica)
DETAILED_ELEMENT	representation_item	510		{representation_item => (geometric_representation_item) (mapped_item)}
name	representation_item. name	510		

Table 10 – Mapping table wireframe_model_3d UoF (G2) (continued)

Application element	AIM element	Source	Rules	Reference path
detailed_element to multi_ language_string (as name)	PATH		154	representation_item multi_language_attribute_item = representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
DETAILED_ GEOMETRIC_MODEL_ ELEMENT	geometric_ representation_item	510		
DETAILED_MODEL_ ELEMENT	representation_item	510		{representation_item => (geometric_representation_item) (mapped_item)}
ELLIPSE	ellipse	510		
GEOMETRIC_MODEL	shape_representation	510	43,187	
description	representation. description	510		shape_representation <= representation representation.description
model_extent	uncertainty_measure_ with_unit	43		shape_representation <= representation => uncertainty_assigned_representation uncertainty_assigned_representation.uncertainty[i] -> uncertainty_measure_with_unit {uncertainty_measure_with_unit.name = 'model extent'} {uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit}
model_id	representation.id	510		shape_representation <= representation representation.id
role	representation.name	510		shape_representation <= representation representation.name {(representation.name) (representation.name = 'design shape')

Table 10 – Mapping table wireframe_model_3d UoF (G2) (continued)

Application element	AIM element	Source	Rules	Reference path
				(representation.name = 'idealized shape')}
geometric_model to template_ instance (as additional_ elements)	PATH			shape_representation <= representation representation.items[i] -> mapped_item
geometric_model to multi_ language_string (as description)	PATH		162	shape_representation multi_language_attribute_item = shape_representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation.description'}
geometric_model to cartesian_ coordinate_space (as is_defined_ in)	PATH			shape_representation => representation representation.context_of_items -> representation_context => geometric_representation_context {geometric_representation_context.coordinate_space_dimension = '3'}
GEOMETRIC_MODEL_ RELATIONSHIP	representation_ relationship	43		
description	representation_ relationship. description	43		
relation_type	representation_ relationship.name	43		
geometric_model_ relationship to multi_language_ string (as description)	PATH		155	representation_relationship multi_language_attribute_item = representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 10 – Mapping table wireframe_model_3d UoF (G2) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_model_ relationship to external_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => externally_defined_representation
geometric_model_ relationship to geometric_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => shape_representation
geometric_model_ relationship to external_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => externally_defined_representation
geometric_model_ relationship to geometric_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => shape_representation
GEOMETRIC_MODEL_ VERSION	shape_representation	510	181,33	{shape_representation version_item = shape_representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
version_id	identification_ assignment.assigned_ id	41	181,33	shape_representation version_item = shape_representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id

Table 10 – Mapping table wireframe_model_3d UoF (G2) (concluded)

Application element	AIM element	Source	Rules	Reference path
HYBRID_GEOMETRIC_MODEL_3D	hybrid_3d_shape_representation	214		hybrid_3d_shape_representation <= shape_representation
hybrid_geometric_model_3d to detailed_geometric_model_element (as element)	PATH			hybrid_3d_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item
HYPERBOLA	hyperbola	510		
LINE	line	510		
OFFSET_CURVE	offset_curve_3d	510		
PARABOLA	parabola	510		
POINT	(cartesian_point) (point_on_curve) (point_replica)	510 510 510		
POINT_ON_CURVE	point_on_curve	510		
POLYLINE	polyline	510		
TRIMMED_CURVE	trimmed_curve	510		
WIREFRAME_MODEL_3D	geometrically_bounded_wireframe_shape_representation	510		geometrically_bounded_wireframe_shape_representation <= shape_representation
wireframe_model_3d to curve_set_3d (as element)	PATH			geometrically_bounded_wireframe_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => geometric_set => geometric_curve_set

Table 11 – Mapping table connected_surface_model UoF (G3)

Application element	AIM element	Source	Rules	Reference path
B_SPLINE_CURVE	b_spline_curve	509		
B_SPLINE_SURFACE	b_spline_surface	509		
CARTESIAN_POINT	cartesian_point	509		
CIRCLE	circle	509		
CONICAL_SURFACE	conical_surface	509		
CONSTRUCTIVE_ GEOMETRIC_ ELEMENT	(axis2_placement_3d) (cartesian_point) (point_on_curve) (point_on_surface) (degenerate_pcurve) (curve) (vertex_point) (advanced_face) (surface) (oriented_edge)	509 509 509 509 509 509 509 509 509 509		
description	representation_item. name	509		((axis2_placement_3d <= placement <=) ((cartesian_point <=) (point_on_curve <=) (point_on_surface <=) (degenerate_pcurve <=) point <=) (surface <=) (curve <=) geometric_representation_item <=) ((vertex_point <= vertex <=) (advanced_face <= face_surface <= face <=) (oriented_edge <= edge <=) topological_representation_item <=) representation_item representation_item.name

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometric_ element to axis_ placement (as element)	PATH			axis2_placement_3d <= placement
constructive_ geometric_ element to curve (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to face (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to oriented_edge_ curve (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to point (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to surface (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to vertex_point (as element)	IDENTICAL MAPPING			
CONSTRUCTIVE_ GEOMETRY	constructive_ geometry_ representation	214		constructive_geometry_representation <= representation {representation.context_of_items ->

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation_context => geometric_representation_context geometric_representation_context.coordinate_space_dimension = '3'}
constructive_ geometry to constructive_ geometric_ element (as element)	PATH			constructive_geometry_representation <= representation representation.items[i] ->> representation_item => (geometric_representation_item => (placement => axis2_placement_3d) (point => cartesian_point) (point => point_on_curve) (point => point_on_surface) (point => degenerate_pcurve) (curve) (surface)) (topological_representation_item => (vertex => vertex_point) (face => face_surface => advanced_face) (edge => oriented_edge))
CONSTRUCTIVE_ GEOMETRY_ ASSOCIATION	constructive_ geometry_ representation_ relationship	214		constructive_geometry_representation_relationship <= representation_relationship
description	representation_ relationship. description	43		constructive_geometry_representation_relationship <= representation_relationship representation_relationship.description

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometry_ association to constructive_ geometry (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => constructive_geometry_representation
constructive_ geometry_ association to geometric_model (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => shape_representation
constructive_ geometry_ association to constructive_ geometry (as constructive_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_2 -> constructive_geometry_representation
constructive_ geometry_ association to multi_language_ string (as description)	PATH		98	constructive_geometry_representation_relationship multi_language_attribute_item = \ constructive_geometry_representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation_relationship.description'}
CURVE	curve	509		
CURVE_ON_SURFACE	(pcurve) (surface_curve)	509 509		
CYLINDRICAL_ SURFACE	cylindrical_surface	509		

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
DETAILED_ELEMENT	representation_item	509		{representation_item => (geometric_representation_item) (topological_representation_item) (mapped_item)}
name	representation_item. name	509		
detailed_element to multi_ language_string (as name)	PATH		154	representation_item multi_language_attribute_item = representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
DETAILED_ GEOMETRIC_MODEL_ ELEMENT	geometric_ representation_item	509		
DETAILED_MODEL_ ELEMENT	representation_item	509		{representation_item => (geometric_representation_item) (mapped_item)}
EDGE_LOOP	edge_loop	509		
edge_loop to oriented_edge_ curve (as edge)	PATH			edge_loop<= path path.edge_list[i] -> oriented_edge
EDGE_TRANSITION	shape_aspect_ transition	214		shape_aspect_transition <= shape_aspect_relationship
transition	shape_aspect_ relationship.name	41		shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.name {(shape_aspect_relationship.name = 'g0') (shape_aspect_relationship.name = 'g1') (shape_aspect_relationship.name = 'g2')}

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
edge_transition to oriented_edge_ curve (as edge_1)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => topological_representation_item => edge => (edge_curve) (oriented_edge)
edge_transition to oriented_edge_ curve (as edge_2)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation.items[i] -> representation_item => topological_representation_item => edge => (edge_curve) (oriented_edge)
edge_transition to vertex_point (as vertex_ joint)	PATH			shape_aspect_transition <= shape_aspect_relationship [shape_aspect_relationship.relating_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] ->] [shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] ->]

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item => topological_representation_item => vertex => vertex_point
ELLIPSE	ellipse	509		
FACE	advanced_face	509		advanced_face <= face_surface
face to face_bound (as bound)	PATH			advanced_face <= face_surface <= face face.bounds[i] -> (face_bound) (face_bound => face_outer_bound)
face to surface (as face_geometry)	PATH			advanced_face <= face_surface face_surface.face_geometry -> surface
FACE_BOUND	(face_bound) (face_outer_bound)	509 509		
outer	face_outer_bound	509		
face_bound to edge_loop (as loop)	PATH			(face_bound) (face_outer_bound <= face_bound) face_bound.bound -> loop => edge_loop
face_bound to vertex_loop (as loop)	PATH			(face_bound) (face_outer_bound <= face_bound) face_bound.bound -> loop => vertex_loop
FACE_TRANSITION	shape_aspect_transition	214		shape_aspect_transition <= shape_aspect_relationship

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
transition	shape_aspect_ relationship.name	41		shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.name {(shape_aspect_relationship.name = 'g0') (shape_aspect_relationship.name = 'g1') (shape_aspect_relationship.name = 'g2')}
face_transition to oriented_edge_ curve (as edge_joint)	PATH			shape_aspect_transition <= shape_aspect_relationship [shape_aspect_relationship.relating_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] ->] [shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] ->]

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item => topological_representation_item => edge => (oriented_edge) (edge_curve)
face_transition to face (as face_1)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect ->shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->representation representation.items[i] ->representation_item => topological_representation_item => face => (oriented_face) (face_surface)
face_transition to face (as face_2)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect ->shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => topological_representation_item => face => (oriented_face) (face_surface)
GEOMETRIC_MODEL	shape_representation	509	43,187	
description	representation. description	509		shape_representation <= representation representation.description
model_extent	uncertainty_measure_ with_unit	43		shape_representation <= representation => uncertainty_assigned_representation uncertainty_assigned_representation.uncertainty[i] -> uncertainty_measure_with_unit {uncertainty_measure_with_unit.name = 'model extent'} {uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit}
model_id	representation.id	509		shape_representation <= representation representation.id
role	representation.name	509		shape_representation <= representation representation.name {(representation.name) (representation.name = 'design shape') (representation.name = 'idealized shape')}

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_model to template_instance (as additional_elements)	PATH			shape_representation <= representation representation.items[i] -> mapped_item
geometric_model to multi_language_string (as description)	PATH		162	shape_representation multi_language_attribute_item = shape_representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation.description'}
geometric_model to cartesian_coordinate_space (as is_defined_in)	PATH			shape_representation => representation representation.context_of_items -> representation_context => geometric_representation_context {geometric_representation_context.coordinate_space_dimension = '3'}
GEOMETRIC_MODEL_RELATIONSHIP	representation_relationship	43		
description	representation_relationship.description	43		
relation_type	representation_relationship.name	43		
geometric_model_relationship to multi_language_string (as description)	PATH		155	representation_relationship multi_language_attribute_item = representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_model_ relationship to external_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => externally_defined_representation
geometric_model_ relationship to geometric_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => shape_representation
geometric_model_ relationship to external_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => externally_defined_representation
geometric_model_ relationship to geometric_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => shape_representation
GEOMETRIC_MODEL_ VERSION	shape_representation	509	181,33	{shape_representation version_item = shape_representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
version_id	identification_ assignment.assigned_ id	41	181,33	{shape_representation version_item = shape_representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
HYBRID_GEOMETRIC_MODEL_3D	hybrid_3d_shape_representation	214		hybrid_3d_shape_representation <= shape_representation
hybrid_geometric_model_3d to detailed_geometric_model_element (as element)	PATH			hybrid_3d_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => (geometric_representation_item) (topological_representation_item)
HYPERBOLA	hyperbola	509		
LINE	line	509		
OFFSET_SURFACE	offset_surface	509		
ORIENTED_EDGE_CURVE	oriented_edge	509		
oriented_edge_curve to vertex_point (as edge_end)	PATH			oriented_edge <= edge edge.edge_end -> vertex => vertex_point
oriented_edge_curve to b_spline_curve (as edge_geometry)	PATH			oriented_edge oriented_edge.edge_element -> edge_curve edge_curve.edge_geometry -> curve => bounded_curve => b_spline_curve
oriented_edge_curve to circle (as edge_geometry)	PATH			oriented_edge oriented_edge.edge_element -> edge_curve edge_curve.edge_geometry -> curve => conic => circle

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
oriented_edge_ curve to curve_ on_surface (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->
edge_curve edge_curve.edge_geometry ->
curve => (pcurve) (surface_curve)
oriented_edge_ curve to ellipse (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->
edge_curve edge_curve.edge_geometry ->
curve => conic => ellipse
oriented_edge_ curve to hyperbola (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->
edge_curve edge_curve.edge_geometry ->
curve => conic => hyperbola
oriented_edge_ curve to line (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->
edge_curve edge_curve.edge_geometry ->
curve => line
oriented_edge_ curve to parabola (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->
edge_curve edge_curve.edge_geometry ->
curve => conic => parabola

Table 11 – Mapping table connected_surface_model UoF (G3) (continued)

Application element	AIM element	Source	Rules	Reference path
oriented_edge_ curve to polyline (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->& edge_curve edge_curve.edge_geometry ->& curve => bounded_curve => polyline
oriented_edge_ curve to vertex_ point (as edge_start)	PATH			oriented_edge <= edge edge.edge_start ->& vertex => vertex_point
PARABOLA	parabola	509		
PLANE	plane	509		
POINT	(cartesian_point) (point_on_curve) (point_on_surface) (degenerate_pcurve)	509 509 509 509		
POINT_ON_CURVE	point_on_curve	509		
POINT_ON_SURFACE	point_on_surface	509		
POLYLINE	polyline	509		
SHELL	(closed_shell) (open_shell)	509 509		
shell to face (as face)	PATH			(closed_shell <=) (open_shell <=) connected_face_set connected_face_set.cfs_faces[i] ->& face => face_surface => advanced_face
SPHERICAL_ SURFACE	spherical_surface	509		
SURFACE	surface	509		
SURFACE_MODEL	manifold_surface_ shape_representation	509		manifold_surface_shape_representation <= shape_representation

Table 11 – Mapping table connected_surface_model UoF (G3) (concluded)

Application element	AIM element	Source	Rules	Reference path
surface_model to shell (as element)	PATH			manifold_surface_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => shell_based_surface_model shell_based_surface_model.sbsm_boundary -> shell (shell = closed_shell closed_shell) (shell = open_shell open_shell)
SURFACE_OF_ LINEAR_EXTRUSION	surface_of_linear_ extrusion	509		
SURFACE_OF_ REVOLUTION	surface_of_ revolution	509		
TOROIDAL_SURFACE	(toroidal_surface) (degenerate_toroidal_ surface)	509 509		
VERTEX_LOOP	vertex_loop	509		
vertex_loop to vertex_point (as loop_vertex)	PATH			vertex_loop vertex_loop.loop_vertex -> vertex => vertex_point
VERTEX_POINT	vertex_point	509		
vertex_point to point (as vertex_ geometry)	PATH			vertex_point vertex_point.vertex_geometry -> point => (cartesian_point) (point_on_curve) (point_on_surface) (degenerate_pcurve)

Table 12 – Mapping table faceted_b_rep_model UoF (G4)

Application element	AIM element	Source	Rules	Reference path
CARTESIAN_POINT	cartesian_point	512		
CONSTRUCTIVE_ GEOMETRIC_ ELEMENT	(axis2_placement_3d) (cartesian_point) (face_surface) (surface) (oriented_edge)	512 512 512 512 512		
description	representation_item. name	512		((axis2_placement_3d <= placement <=) (cartesian_point <= point <=) (surface <=) geometric_representation_item <=) ((face_surface <= face <=) (oriented_edge <= edge <=) topological_representation_item <=) representation_item representation_item.name
constructive_ geometric_ element to axis_ placement (as element)	PATH			axis2_placement_3d <= placement
constructive_ geometric_ element to point (as element)	IDENTICAL MAPPING			
CONSTRUCTIVE_ GEOMETRY	constructive_ geometry_ representation	214		constructive_geometry_representation <= representation {representation.context_of_items -> representation_context => geometric_representation_context geometric_representation_context.coordinate_space_dimension = '3'}

Table 12 – Mapping table faceted_b_rep_model UoF (G4) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometry to constructive_ geometric_ element (as element)	PATH			constructive_geometry_representation <= representation representation.items[i] -> representation_item => (geometric_representation_item => (placement => axis2_placement_3d) (point => cartesian_point) (surface)) (topological_representation_item => (face => face_surface) (edge => oriented_edge))
CONSTRUCTIVE_ GEOMETRY_ ASSOCIATION	constructive_ geometry_ representation_ relationship	214		constructive_geometry_representation_relationship <= representation_relationship
description	representation_ relationship. description	43		constructive_geometry_representation_relationship <= representation_relationship representation_relationship.description
constructive_ geometry_ association to constructive_ geometry (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => constructive_geometry_representation
constructive_ geometry_ association to geometric_model (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => shape_representation

Table 12 – Mapping table faceted_b_rep_model UoF (G4) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometry_ association to constructive_ geometry (as constructive_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_2 -> constructive_geometry_representation
constructive_ geometry_ association to multi_language_ string (as description)	PATH		98	constructive_geometry_representation_relationship multi_language_attribute_item = \ constructive_geometry_representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation_relationship.description'}
DETAILED_ELEMENT	representation_item	512		{representation_item => (geometric_representation_item) (topological_representation_item) (mapped_item)}
name	representation_item. name	512		
detailed_element to multi_ language_string (as name)	PATH		154	representation_item multi_language_attribute_item = representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
DETAILED_ GEOMETRIC_MODEL_ ELEMENT	geometric_ representation_item	512		

Table 12 – Mapping table faceted_b_rep_model UoF (G4) (continued)

Application element	AIM element	Source	Rules	Reference path
DETAILED_MODEL_ ELEMENT	representation_item	512		{representation_item => (geometric_representation_item) (mapped_item)}
FACETED_B_REP	(faceted_brep) (brep_with_voids)	512 512		
faceted_b_rep to faceted_closed_ shell (as outer)	PATH			(faceted_brep <=) (brep_with_voids <=) manifold_solid_brep manifold_solid_brep.outer -> closed_shell
faceted_b_rep to faceted_closed_ shell (as void)	PATH			brep_with_voids brep_with_voids.voids[i] -> oriented_closed_shell <= closed_shell
FACETED_B_REP_ MODEL	faceted_brep_shape_ representation	512		faceted_brep_shape_representation <= shape_representation
faceted_b_rep_ model to faceted_ b_rep (as element)	PATH			faceted_brep_shape_representation <= shape_representation representation representation.items[i] -> representation_item => geometric_representation_item => solid_model => manifold_solid_brep => (faceted_brep) (brep_with_voids)
FACETED_CLOSED_ SHELL	closed_shell	512		
faceted_closed_ shell to faceted_ face (as shell_ boundary)	PATH			closed_shell <= connected_face_set connected_face_set.cfs_faces[i] -> face => face_surface
FACETED_FACE	face_surface	512		

Table 12 – Mapping table faceted_b_rep_model UoF (G4) (continued)

Application element	AIM element	Source	Rules	Reference path
faceted_face to poly_loop (as bound)	PATH			face_surface <= face face.bounds[i] -> (face_bound => face_outer_bound) (face_bound) face_bound.bound -> loop => poly_loop
faceted_face to plane (as face_geometry)	PATH			face_surface face_surface.face_geometry -> surface => elementary_surface => plane
GEOMETRIC_MODEL	shape_representation	512	43,187	
description	representation. description	512		shape_representation <= representation representation.description
model_extent	uncertainty_measure_with_unit	43		shape_representation <= representation => uncertainty_assigned_representation uncertainty_assigned_representation.uncertainty[i] -> uncertainty_measure_with_unit {uncertainty_measure_with_unit.name = 'model extent'} {uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit}
model_id	representation.id	512		shape_representation <= representation representation.id
role	representation.name	512		shape_representation <= representation representation.name {(representation.name) (representation.name = 'design shape') (representation.name = 'idealized shape')}

Table 12 – Mapping table faceted_b_rep_model UoF (G4) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_model to template_ instance (as additional_ elements)	PATH			shape_representation <= representation representation.items[i] -> mapped_item
geometric_model to multi_ language_string (as description)	PATH		162	shape_representation multi_language_attribute_item = shape_representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation.description'}
geometric_model to cartesian_ coordinate_space (as is_defined_ in)	PATH			shape_representation => representation representation.context_of_items -> representation_context => geometric_representation_context {geometric_representation_context.coordinate_space_dimension = '3'}
GEOMETRIC_MODEL_ RELATIONSHIP	representation_ relationship	43		
description	representation_ relationship. description	43		
relation_type	representation_ relationship.name	43		
geometric_model_ relationship to multi_language_ string (as description)	PATH		155	representation_relationship multi_language_attribute_item = representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 12 – Mapping table faceted_b_rep_model UoF (G4) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_model_ relationship to external_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => externally_defined_representation
geometric_model_ relationship to geometric_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => shape_representation
geometric_model_ relationship to external_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => externally_defined_representation
geometric_model_ relationship to geometric_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => shape_representation
GEOMETRIC_MODEL_ VERSION	representation	512	180,33	{representation version_item = representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
version_id	identification_ assignment.assigned_ id	41	181,33	{shape_representation version_item = shape_representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}

Table 12 – Mapping table faceted_b_rep_model UoF (G4) (concluded)

Application element	AIM element	Source	Rules	Reference path
HYBRID_GEOMETRIC_MODEL_3D	hybrid_3d_shape_representation	214		hybrid_3d_shape_representation <= shape_representation
hybrid_geometric_model_3d to detailed_geometric_model_element (as element)	PATH			hybrid_3d_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => (geometric_representation_item) (topological_representation_item)
PLANE	plane	512		
POINT	cartesian_point	512		
POLY_LOOP	poly_loop	512		
outer	face_outer_bound	512		poly_loop <= loop <- face_bound.bound face_bound => face_outer_bound
poly_loop to cartesian_point (as polygon)	PATH			poly_loop poly_loop.polygon[i] -> cartesian_point

Table 13 – Mapping table b_rep_model UoF (G5)

Application element	AIM element	Source	Rules	Reference path
B_REP	(|manifold_solid_ brep|) (brep_with_voids)	514 514		
b_rep to closed_ shell (as outer)	PATH			(|manifold_solid_brep|) (brep_with_voids <= manifold_solid_brep) manifold_solid_brep.outer ->|closed_shell|
b_rep to closed_ shell (as void)	PATH			brep_with_voids brep_with_voids.voids[i] ->oriented_closed_shell <= closed_shell
B_REP_MODEL	advanced_brep_shape_ representation	514		advanced_brep_shape_representation <= shape_representation
b_rep_model to b_ rep (as element)	PATH			advanced_brep_shape_representation <= shape_representation <= representation representation.items[i] ->representation_item => geometric_representation_item => solid_model => (|manifold_solid_brep|) (manifold_solid_brep => brep_with_voids)
B_SPLINE_CURVE	b_spline_curve	514		
B_SPLINE_SURFACE	b_spline_surface	514		
CARTESIAN_POINT	cartesian_point	514		
CIRCLE	circle	514		
CLOSED_SHELL	(|closed_shell|) (oriented_closed_ shell)	514 514		
closed_shell to face (as shell_ boundary)	PATH			(|closed_shell <=|) (oriented_closed_shell <= closed_shell <=) connected_face_set connected_face_set.cfs_faces[i] ->

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
				face => face_surface => advanced_face
CONICAL_SURFACE	conical_surface	514		
CONSTRUCTIVE_ GEOMETRIC_ ELEMENT	(axis2_placement_3d) (cartesian_point) (curve) (vertex_point) (advanced_face) (surface) (oriented_edge)	514 514 514 514 514 514 514		
description	representation_item. name	514		((axis2_placement_3d <= placement <=) (cartesian_point <= point <=) (surface <=) (curve <=) geometric_representation_item <=) ((vertex_point <= vertex <=) (advanced_face <= face_surface <= face <=) (oriented_edge <= edge <=) topological_representation_item <=) representation_item representation_item.name
constructive_ geometric_ element to axis_ placement (as element)	PATH			axis2_placement_3d <= placement

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometric_ element to curve (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to face (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to oriented_edge_ curve (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to point (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to surface (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to vertex_point (as element)	IDENTICAL MAPPING			
CONSTRUCTIVE_ GEOMETRY	constructive_ geometry_ representation	214		constructive_geometry_representation <= representation {representation.context_of_items -> representation_context => geometric_representation_context geometric_representation_context.coordinate_space_dimension = '3'}

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometry to constructive_ geometric_ element (as element)	PATH			constructive_geometry_representation <= representation representation.items[i] -> representation_item => (geometric_representation_item (placement => axis2_placement_3d) (point => cartesian_point) (point => point_on_curve) (point => point_on_surface) (point => degenerate_pcurve) (curve) (surface)) (topological_representation_item (vertex => vertex_point) (face => face_surface => advanced_face) (edge => oriented_edge))
CONSTRUCTIVE_ GEOMETRY_ ASSOCIATION	constructive_ geometry_ representation_ relationship	214		constructive_geometry_representation_relationship <= representation_relationship
description	representation_ relationship. description	43		constructive_geometry_representation_relationship <= representation_relationship representation_relationship.description

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometry_ association to constructive_ geometry (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => constructive_geometry_representation
constructive_ geometry_ association to geometric_model (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => shape_representation
constructive_ geometry_ association to constructive_ geometry (as constructive_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_2 -> constructive_geometry_representation
constructive_ geometry_ association to multi_language_ string (as description)	PATH		98	constructive_geometry_representation_relationship multi_language_attribute_item = \ constructive_geometry_representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation_relationship.description'}
CURVE	curve	514		
CURVE_ON_SURFACE	(pcurve) (surface_curve)	514 514		
CYLINDRICAL_ SURFACE	cylindrical_surface	514		

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
DETAILED_ELEMENT	representation_item	514		{representation_item => (geometric_representation_item) (topological_representation_item) (mapped_item)}
name	representation_item. name	514		
detailed_element to multi_ language_string (as name)	PATH		154	representation_item multi_language_attribute_item = representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
DETAILED_ GEOMETRIC_MODEL_ ELEMENT	geometric_ representation_item	514		
DETAILED_MODEL_ ELEMENT	representation_item	514		{representation_item => (geometric_representation_item) (mapped_item)}
EDGE_LOOP	edge_loop	514		
edge_loop to oriented_edge_ curve (as edge)	PATH			edge_loop<= path path.edge_list[i] -> oriented_edge
EDGE_TRANSITION	shape_aspect_ transition	214		shape_aspect_transition <= shape_aspect_relationship
transition	shape_aspect_ relationship.name	41		shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.name
edge_transition to oriented_edge_ curve (as edge_1)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => topological_representation_item => edge => (edge_curve) (oriented_edge)
edge_transition to oriented_edge_ curve (as edge_2)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => topological_representation_item => edge => (edge_curve) (oriented_edge)

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
edge_transition to vertex_point (as vertex_ joint)	PATH			shape_aspect_transition <= shape_aspect_relationship [shape_aspect_relationship.relating_shape_aspect ->] shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->] representation representation.items[i] ->] [shape_aspect_relationship.related_shape_aspect ->] shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->] representation representation.items[i] ->] representation_item => topological_representation_item => vertex => vertex_point
ELLIPSE	ellipse	514		

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
FACE	advanced_face	514		advanced_face <= face_surface
face to face_bound (as bound)	PATH			advanced_face <= face_surface <= face face.bounds[i] -> (face_bound) (face_bound => face_outer_bound)
face to surface (as face_geometry)	PATH			advanced_face <= face_surface face_surface.face_geometry -> surface
FACE_BOUND	(face_bound) (face_outer_bound)	514 514		
outer	face_outer_bound	514		
face_bound to edge_loop (as loop)	PATH			(face_bound) (face_outer_bound <= face_bound) face_bound.bound -> loop => edge_loop
face_bound to vertex_loop (as loop)	PATH			(face_bound) (face_outer_bound <= face_bound) face_bound.bound -> loop => vertex_loop
FACE_TRANSITION	shape_aspect_transition	214		shape_aspect_transition <= shape_aspect_relationship
transition	shape_aspect_relationship.name	41		shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.name

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
face_transition to oriented_edge_ curve (as edge_joint)	PATH			shape_aspect_transition <= shape_aspect_relationship [shape_aspect_relationship.relating_shape_aspect ->] shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->] representation representation.items[i] ->] [shape_aspect_relationship.related_shape_aspect ->] shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->] representation representation.items[i] ->] representation_item => topological_representation_item => edge => (oriented_edge) (edge_curve)

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
face_transition to face (as face_1)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect ->
shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->
representation representation.items[i] ->
representation_item => topological_representation_item => face => (oriented_face) (face_surface)
face_transition to face (as face_2)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect ->
shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->
representation

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation.items[i] -> representation_item => topological_representation_item => face => (oriented_face) (face_surface)
GEOMETRIC_MODEL	shape_representation	514	43,187	
description	representation. description	514		shape_representation <= representation representation.description
model_extent	uncertainty_measure_ with_unit	43		shape_representation <= representation => uncertainty_assigned_representation uncertainty_assigned_representation.uncertainty[i] -> uncertainty_measure_with_unit {uncertainty_measure_with_unit.name = 'model extent'} {uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit}
model_id	representation.id	514		shape_representation <= representation representation.id
role	representation.name	514		shape_representation <= representation representation.name {(representation.name) (representation.name = 'design shape') (representation.name = 'idealized shape')}
geometric_model to template_ instance (as additional_ elements)	PATH			shape_representation <= representation representation.items[i] -> mapped_item

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_model to multi_ language_string (as description)	PATH		162	shape_representation multi_language_attribute_item = shape_representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation.description'}
geometric_model to cartesian_ coordinate_space (as is_defined_ in)	PATH			shape_representation => representation representation.context_of_items -> representation_context => geometric_representation_context {geometric_representation_context.coordinate_space_dimension = '3'}
GEOMETRIC_MODEL_ RELATIONSHIP	representation_ relationship	43		
description	representation_ relationship. description	43		
relation_type	representation_ relationship.name	43		
geometric_model_ relationship to multi_language_ string (as description)	PATH		155	representation_relationship multi_language_attribute_item = representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
geometric_model_ relationship to external_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => externally_defined_representation

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_model_ relationship to geometric_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => shape_representation
geometric_model_ relationship to external_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => externally_defined_representation
geometric_model_ relationship to geometric_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => shape_representation
GEOMETRIC_MODEL_ VERSION	representation	514	180,33	{representation version_item = representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
version_id	identification_ assignment.assigned_ id	41	181,33	{shape_representation version_item = shape_representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
HYBRID_GEOMETRIC_ MODEL_3D	hybrid_3d_shape_ representation	214		hybrid_3d_shape_representation <= shape_representation

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
hybrid_geometric_model_3d to detailed_geometric_model_element (as element)	PATH			hybrid_3d_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => (geometric_representation_item) (topological_representation_item)
HYPERBOLA	hyperbola	514		
LINE	line	514		
ORIENTED_EDGE_CURVE	oriented_edge	514		
oriented_edge_curve to vertex_point (as edge_end)	PATH			oriented_edge <= edge edge.edge_end -> vertex => vertex_point
oriented_edge_curve to b_spline_curve (as edge_geometry)	PATH			oriented_edge oriented_edge.edge_element -> edge_curve edge_curve.edge_geometry -> curve => bounded_curve => b_spline_curve
oriented_edge_curve to circle (as edge_geometry)	PATH			oriented_edge oriented_edge.edge_element -> edge_curve edge_curve.edge_geometry -> curve => conic => circle
oriented_edge_curve to curve_on_surface (as edge_geometry)	PATH			oriented_edge oriented_edge.edge_element -> edge_curve edge_curve.edge_geometry -> curve => (pcurve)

Table 13 – Mapping table b_rep_model UoF (G5) (continued)

Application element	AIM element	Source	Rules	Reference path
				(surface_curve)
oriented_edge_ curve to ellipse (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->& edge_curve edge_curve.edge_geometry ->& curve => conic => ellipse
oriented_edge_ curve to hyperbola (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->& edge_curve edge_curve.edge_geometry ->& curve => conic => hyperbola
oriented_edge_ curve to line (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->& edge_curve edge_curve.edge_geometry ->& curve => line
oriented_edge_ curve to parabola (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->& edge_curve edge_curve.edge_geometry ->& curve => conic => parabola
oriented_edge_ curve to polyline (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->& edge_curve edge_curve.edge_geometry ->& curve => bounded_curve => polyline

Table 13 – Mapping table b_rep_model UoF (G5) (concluded)

Application element	AIM element	Source	Rules	Reference path
oriented_edge_ curve to vertex_ point (as edge_start)	PATH			oriented_edge <= edge edge.edge_start -> vertex => vertex_point
PARABOLA	parabola	514		
PLANE	plane	514		
POINT	cartesian_point	514		
POLYLINE	polyline	514		
SPHERICAL_ SURFACE	spherical_surface	514		
SURFACE	surface	514		
SURFACE_OF_ LINEAR_EXTRUSION	surface_of_linear_ extrusion	514		
SURFACE_OF_ REVOLUTION	surface_of_ revolution	514		
TOROIDAL_SURFACE	toroidal_surface	514		
VERTEX_LOOP	vertex_loop	514		
vertex_loop to vertex_point (as loop_vertex)	PATH			vertex_loop vertex_loop.loop_vertex -> vertex => vertex_point
VERTEX_POINT	vertex_point	514		
vertex_point to point (as vertex_ geometry)	PATH			vertex_point vertex_point.vertex_geometry -> point => cartesian_point

Table 14 – Mapping table compound_model UoF (G6)

Application element	AIM element	Source	Rules	Reference path
B_SPLINE_CURVE	b_spline_curve	501, 508, 42		
B_SPLINE_SURFACE	b_spline_surface	501, 508, 42		
CARTESIAN_POINT	cartesian_point	501, 508, 42		
CIRCLE	circle	501, 508, 42		
COMPOUND_MODEL	(compound_shape_ representation) (edge_based_ wireframe_shape_ representation) (non_manifold_ surface_shape_ representation)	214 501 508		(compound_shape_representation <=) (edge_based_wireframe_shape_representation <=) (non_manifold_surface_shape_representation <=) shape_representation
compound_model to topologically_ connected_set (as element)	PATH			((compound_shape_representation <=) (edge_based_wireframe_shape_representation <=) shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => edge_based_wireframe_model) ((compound_shape_representation <=) (non_manifold_surface_shape_representation <=) shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item =>

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
				face_based_surface_model)
CONICAL_SURFACE	conical_surface	508, 42		
CONSTRUCTIVE_ GEOMETRIC_ ELEMENT	(axis2_placement_3d)	501, 508, 42		
	(cartesian_point)	501, 508, 42		
	(point_on_curve)	508, 42		
	(point_on_surface)	508, 42		
	(degenerate_pcurve)	508, 42		
	(curve)	501, 508, 42		
	(vertex_point)	501, 508, 42		
	(advanced_face)	508		
	(surface)	508, 42		
	(oriented_edge)	508, 42		
description	representation_item. name	43		((axis2_placement_3d <= placement <=) ((cartesian_point <=) (point_on_curve <=) (point_on_surface <=) (degenerate_pcurve <=) point <=) (surface <=) (curve <=) geometric_representation_item <=)

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
				((vertex_point <= vertex <=) (advanced_face <= face_surface <= face <=) (oriented_edge <= edge <=) topological_representation_item <=) representation_item representation_item.name
constructive_ geometric_ element to axis_ placement (as element)	PATH			axis2_placement_3d <= placement
constructive_ geometric_ element to curve (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to face (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to oriented_edge_ curve (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to point (as element)	IDENTICAL MAPPING			

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometric_ element to surface (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to vertex_point (as element)	IDENTICAL MAPPING			
CONSTRUCTIVE_ GEOMETRY	constructive_ geometry_ representation	214		constructive_geometry_representation <= representation {representation.context_of_items -> representation_context => geometric_representation_context geometric_representation_context.coordinate_space_dimension = '3'}
constructive_ geometry to constructive_ geometric_ element (as element)	PATH			constructive_geometry_representation <= representation representation.items[i] -> representation_item => (geometric_representation_item (placement => axis2_placement_3d) (point => cartesian_point) (point => point_on_curve) (point => point_on_surface) (point => degenerate_pcurve) (curve) (surface)) (topological_representation_item (vertex => vertex_point)

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
				(face => face_surface => advanced_face) (edge => oriented_edge))
CONSTRUCTIVE_ GEOMETRY_ ASSOCIATION	constructive_ geometry_ representation_ relationship	214		constructive_geometry_representation_relationship <= representation_relationship
description	representation_ relationship. description	43		constructive_geometry_representation_relationship <= representation_relationship representation_relationship.description
constructive_ geometry_ association to constructive_ geometry (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => constructive_geometry_representation
constructive_ geometry_ association to geometric_model (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => shape_representation
constructive_ geometry_ association to constructive_ geometry (as constructive_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_2 -> constructive_geometry_representation

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometry_ association to multi_language_ string (as description)	PATH		98	constructive_geometry_representation_relationship multi_language_attribute_item = \ constructive_geometry_representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation_relationship.description'}
CURVE	curve	501, 508, 42		
CURVE_ON_SURFACE	(pcurve) (surface_curve)	508, 42 508, 42		
CYLINDRICAL_ SURFACE	cylindrical_surface	508, 42		
DETAILED_ELEMENT	representation_item	501, 508, 43		{representation_item => (geometric_representation_item) (topological_representation_item) (mapped_item)}
name	representation_item. name	43		
detailed_element to multi_ language_string (as name)	PATH		154	representation_item multi_language_attribute_item = representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
DETAILED_ GEOMETRIC_MODEL_ ELEMENT	geometric_ representation_item	42		
DETAILED_MODEL_ ELEMENT	representation_item	501, 508, 43		{representation_item => (geometric_representation_item) (mapped_item)}
EDGE_LOOP	edge_loop	508, 42		
edge_loop to oriented_edge_ curve (as edge)	PATH			edge_loop<= path path.edge_list[i] -> oriented_edge
EDGE_TRANSITION	shape_aspect_ transition	214		shape_aspect_transition <= shape_aspect_relationship
transition	shape_aspect_ relationship.name	41		shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.name
edge_transition to oriented_edge_ curve (as edge_1)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => topological_representation_item => edge =>

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
				(edge_curve) (oriented_edge)
edge_transition to oriented_edge_ curve (as edge_2)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect ->
shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->
representation representation.items[i] ->
representation_item => topological_representation_item => edge => (edge_curve) (oriented_edge)
edge_transition to vertex_point (as vertex_ joint)	PATH			shape_aspect_transition <= shape_aspect_relationship [shape_aspect_relationship.relating_shape_aspect ->
shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation}

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_representation -> representation representation.items[i] ->] [shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] ->] representation_item => topological_representation_item => vertex => vertex_point
ELLIPSE	ellipse	501, 508, 42		
FACE	(face_surface) (oriented_face)	42 42		
face to face_ bound (as bound)	PATH			(face_surface <=) (oriented_face <=) face face.bounds[i] -> (face_bound) (face_bound => face_outer_bound)

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
face to surface (as face_ geometry)	PATH			(face_surface) (oriented_face oriented_face.face_element -> face => face_surface) face_surface.face_geometry -> surface
FACE_BOUND	(face_bound) (face_outer_bound)	508, 42 508, 42		
outer	face_outer_bound	508, 42		
face_bound to edge_loop (as loop)	PATH			(face_bound) (face_outer_bound <= face_bound) face_bound.bound -> loop => edge_loop
face_bound to vertex_loop (as loop)	PATH			(face_bound) (face_outer_bound <= face_bound) face_bound.bound -> loop => vertex_loop
FACE_TRANSITION	shape_aspect_ transition	214		shape_aspect_transition <= shape_aspect_relationship
transition	shape_aspect_ relationship.name	41		shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.name
face_transition to oriented_edge_ curve (as edge_joint)	PATH			shape_aspect_transition <= shape_aspect_relationship [shape_aspect_relationship.relating_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] ->] [shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] ->] representation_item => topological_representation_item => edge => (oriented_edge) (edge_curve)
face_transition to face (as face_1)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <-

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => topological_representation_item => face => (oriented_face) (face_surface)
face_transition to face (as face_2)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => topological_representation_item => face => (oriented_face) (face_surface)

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
GEOMETRIC_MODEL	shape_representation	501, 508	43,187	
description	representation. description	43		shape_representation <= representation representation.description
model_extent	uncertainty_measure_ with_unit	43		shape_representation <= representation => uncertainty_assigned_representation uncertainty_assigned_representation.uncertainty[i] -> uncertainty_measure_with_unit {uncertainty_measure_with_unit.name = 'model extent'} {uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit}
model_id	representation.id	43		shape_representation <= representation representation.id
role	representation.name	43		shape_representation <= representation representation.name {(representation.name) (representation.name = 'design shape') (representation.name = 'idealized shape')}
geometric_model to template_ instance (as additional_ elements)	PATH			shape_representation <= representation representation.items[i] -> mapped_item
geometric_model to multi_ language_string (as description)	PATH		162	shape_representation multi_language_attribute_item = shape_representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
				'representation.description'}
geometric_model to cartesian_ coordinate_space (as is_defined_ in)	PATH			shape_representation => representation representation.context_of_items -> representation_context => geometric_representation_context {geometric_representation_context.coordinate_space_dimension = '3'}
GEOMETRIC_MODEL_ RELATIONSHIP	representation_ relationship	43		
description	representation_ relationship. description	43		
relation_type	representation_ relationship.name	43		
geometric_model_ relationship to multi_language_ string (as description)	PATH		155	representation_relationship multi_language_attribute_item = representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
geometric_model_ relationship to external_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => externally_defined_representation
geometric_model_ relationship to geometric_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => shape_representation
geometric_model_ relationship to external_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => externally_defined_representation

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_model_ relationship to geometric_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => shape_representation
GEOMETRIC_MODEL_ VERSION	representation	501, 508	180,33	{representation version_item = representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
version_id	identification_ assignment.assigned_ id	41	181,33	{shape_representation version_item = shape_representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
HYBRID_GEOMETRIC_ MODEL_3D	hybrid_3d_shape_ representation	214		hybrid_3d_shape_representation <= shape_representation
hybrid_geometric_ model_3d to detailed_ geometric_model_ element (as element)	PATH			hybrid_3d_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => (geometric_representation_item) (topological_representation_item)
HYPERBOLA	hyperbola	501, 508, 42		
LINE	line	501, 508,		

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
		42		
OFFSET_SURFACE	offset_surface	508, 42		
ORIENTED_EDGE_ CURVE	oriented_edge	508, 42		
oriented_edge_ curve to vertex_ point (as edge_end)	PATH			oriented_edge <= edge edge.edge_end -> vertex => vertex_point
oriented_edge_ curve to b_ spline_curve (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element -> edge_curve edge_curve.edge_geometry -> curve => bounded_curve => b_spline_curve
oriented_edge_ curve to circle (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element -> edge_curve edge_curve.edge_geometry -> curve => conic => circle
oriented_edge_ curve to curve_ on_surface (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element -> edge_curve edge_curve.edge_geometry -> curve => (pcurve) (surface_curve)
oriented_edge_ curve to ellipse (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element -> edge_curve edge_curve.edge_geometry -> curve =>

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
				conic => ellipse
oriented_edge_ curve to hyperbola (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->> edge_curve edge_curve.edge_geometry ->> curve => conic => hyperbola
oriented_edge_ curve to line (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->> edge_curve edge_curve.edge_geometry ->> curve => line
oriented_edge_ curve to parabola (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->> edge_curve edge_curve.edge_geometry ->> curve => conic => parabola
oriented_edge_ curve to polyline (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->> edge_curve edge_curve.edge_geometry ->> curve => bounded_curve => polyline
oriented_edge_ curve to vertex_ point (as edge_start)	PATH			oriented_edge <= edge edge.edge_start ->> vertex => vertex_point
PARABOLA	parabola	501, 508,		

Table 14 – Mapping table compound_model UoF (G6) (continued)

Application element	AIM element	Source	Rules	Reference path
		42		
PLANE	plane	508, 42		
POINT	(cartesian_point) (point_on_curve) (point_on_surface) (degenerate_pcurve)	501, 508, 42 508, 42 508, 42 508, 42		
POINT_ON_CURVE	point_on_curve	508, 42		
POINT_ON_SURFACE	point_on_surface	508, 42		
POLYLINE	polyline	501, 508, 42		
SPHERICAL_ SURFACE	spherical_surface	508, 42		
SURFACE	surface	508, 42		
SURFACE_OF_ LINEAR_EXTRUSION	surface_of_linear_ extrusion	514		
SURFACE_OF_ REVOLUTION	surface_of_ revolution	514		
TOPOLOGICALLY_ CONNECTED_SET	(edge_based_ wireframe_model) (face_based_surface_ model)	42 42		
topologically_ connected_set to face (as element)	PATH			face_based_surface_model face_based_surface_model.fbsm_faces ->
connected_face_set connected_face_set.cfs_faces ->
face =>

Table 14 – Mapping table compound_model UoF (G6) (concluded)

Application element	AIM element	Source	Rules	Reference path
				(face_surface) (oriented_face)
topologically_ connected_set to oriented_edge_ curve (as element)	PATH			edge_based_wireframe_model edge_based_wireframe_model.ebwm_boundary ->>>>

Table 15 – Mapping table csg_model UoF (G7)

Application element	AIM element	Source	Rules	Reference path
B_REP	(|manifold_solid_ brep|) (brep_with_voids)	515 515		
b_rep to closed_ shell (as outer)	PATH			(|manifold_solid_brep|) (brep_with_voids <= manifold_solid_brep) manifold_solid_brep.outer -> |closed_shell|
b_rep to closed_ shell (as void)	PATH			brep_with_voids brep_with_voids.voids[i] -> oriented_closed_shell <= closed_shell
B_REP_MODEL	advanced_brep_shape_ representation	515		advanced_brep_shape_representation <= shape_representation
b_rep_model to b_ rep (as element)	PATH			advanced_brep_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => solid_model => (|manifold_solid_brep|) (manifold_solid_brep => brep_with_voids)
B_SPLINE_CURVE	b_spline_curve	515		
B_SPLINE_SURFACE	b_spline_surface	515		
BLOCK	block	515		
BOOLEAN_RESULT	boolean_result	515		
boolean_ operation	boolean_result. operator	515		
boolean_result to b_rep_model (as first_ operand)	PATH			boolean_result boolean_result.first_operand -> boolean_operand = solid_model solid_model => (|manifold_solid_brep|) (manifold_solid_brep =>

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
				brep_with_voids) (|manifold_solid_brep| <=) (brep_with_voids <= manifold_solid_brep <=) solid_model <= geometric_representation_item <= representation_item <- representation.items[i] representation => shape_representation => advanced_brep_shape_representation
boolean_result to boolean_ result (as first_ operand)	PATH			boolean_result boolean_result.first_operand -> boolean_operand = boolean_result boolean_result
boolean_result to csg_primitive (as first_ operand)	PATH			boolean_result boolean_result.first_operand -> boolean_operand = csg_primitive (csg_primitive = sphere sphere) (csg_primitive = block block) (csg_primitive = right_angular_wedge right_angular_wedge) (csg_primitive = torus torus) (csg_primitive = right_circular_cone right_circular_cone) (csg_primitive = right_circular_cylinder right_circular_cylinder)

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
boolean_result to faceted_b_rep_ model (as first_ operand)	PATH			boolean_result boolean_result.first_operand -> boolean_operand = solid_model solid_model => manifold_solid_brep => (faceted_brep <=) (brep_with_voids <=) manifold_solid_brep <= solid_model <= geometric_representation_item <= representation_item <- representation.items[i] representation => shape_representation => faceted_brep_shape_representation
boolean_result to half_space_ solid (as first_ operand)	PATH			boolean_result boolean_result.first_operand -> boolean_operand = half_space_solid half_space_solid
boolean_result to solid_replica (as first_ operand)	PATH			boolean_result boolean_result.first_operand -> boolean_operand = solid_model solid_model => solid_replica
boolean_result to swept_face_ solid (as first_ operand)	PATH			boolean_result boolean_result.first_operand -> boolean_operand = solid_model solid_model => swept_face_solid => (revolved_face_solid) (extruded_face_solid)

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
boolean_result to b_rep_model (as second_ operand)	PATH			boolean_result boolean_result.second_operand -> boolean_operand = solid_model solid_model => (|manifold_solid_brep|) (manifold_solid_brep => brep_with_voids) (|manifold_solid_brep| <=) (brep_with_voids <= manifold_solid_brep <=) solid_model <= geometric_representation_item <= representation_item <- representation.items[i] representation => shape_representation => advanced_brep_shape_representation
boolean_result to boolean_ result (as second_ operand)	PATH			boolean_result boolean_result.second_operand -> boolean_operand = boolean_result boolean_result
boolean_result to csg_primitive (as second_ operand)	PATH			boolean_result boolean_result.first_operand -> boolean_operand = csg_primitive (csg_primitive = sphere sphere) (csg_primitive = block block) (csg_primitive = right_angular_wedge right_angular_wedge) (csg_primitive = torus torus) (csg_primitive = right_circular_cone right_circular_cone) (csg_primitive = right_circular_cylinder

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
				right_circular_cylinder)
boolean_result to faceted_b_rep_ model (as second_ operand)	PATH			boolean_result boolean_result.first_operand -> boolean_operand = solid_model solid_model => manifold_solid_brep => (faceted_brep <=) (brep_with_voids <=) manifold_solid_brep <= solid_model <= geometric_representation_item <= representation_item <- representation.items[i] representation => shape_representation => faceted_brep_shape_representation
boolean_result to half_space_ solid (as second_ operand)	PATH			boolean_result boolean_result.second_operand -> boolean_operand = half_space_solid half_space_solid
boolean_result to solid_replica (as second_ operand)	PATH			boolean_result boolean_result.second_operand -> boolean_operand = solid_model solid_model => solid_replica
boolean_result to swept_face_ solid (as second_ operand)	PATH			boolean_result boolean_result.second_operand -> boolean_operand = solid_model solid_model => swept_face_solid => (revolved_face_solid) (extruded_face_solid)
CARTESIAN_POINT	cartesian_point	515		
CIRCLE	circle	515		

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
CLOSED_SHELL	(|closed_shell|) (oriented_closed_shell)	515 515		
closed_shell to face (as shell_boundary)	PATH			(|closed_shell <=|) (oriented_closed_shell <= closed_shell <=) connected_face_set connected_face_set.cfs_faces[i] -> face => (face_surface) (oriented_face)
CONICAL_SURFACE	conical_surface	515		
CONSTRUCTIVE_GEOMETRIC_ELEMENT	(axis2_placement_3d) (cartesian_point) (curve) (vertex_point) (face_surface) (oriented_face) (surface) (oriented_edge)	515 515 515 515 515 515 515 515		
description	representation_item.name	515		((axis2_placement_3d <= placement <=) ((cartesian_point <=) (point_on_curve <=) (point_on_surface <=) (degenerate_pcurve <=) point <=) (surface <=) (curve <=) geometric_representation_item <=) ((vertex_point <= vertex <=) ((face_surface <=) (oriented_face <=) face <=) (oriented_edge <=

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
				edge <=) topological_representation_item <=) representation_item representation_item.name
constructive_ geometric_ element to axis_ placement (as element)	PATH			axis2_placement_3d <= placement
constructive_ geometric_ element to curve (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to face (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to oriented_edge_ curve (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to point (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to surface (as element)	IDENTICAL MAPPING			

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometric_ element to vertex_point (as element)	IDENTICAL MAPPING			
CONSTRUCTIVE_ GEOMETRY	constructive_ geometry_ representation	214		constructive_geometry_representation <= representation {representation.context_of_items -> representation_context => geometric_representation_context geometric_representation_context.coordinate_space_dimension = '3'}
constructive_ geometry to constructive_ geometric_ element (as element)	PATH			constructive_geometry_representation <= representation representation.items[i] -> representation_item => (geometric_representation_item (placement => axis2_placement_3d) (point => cartesian_point) (point => point_on_curve) (point => point_on_surface) (point => degenerate_pcurve) (curve) (surface)) (topological_representation_item (vertex => vertex_point) (face => (face_surface) (oriented_face)) (edge => oriented_edge))

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
CONSTRUCTIVE_ GEOMETRY_ ASSOCIATION	constructive_ geometry_ representation_ relationship	214		constructive_geometry_representation_relationship <= representation_relationship
description	representation_ relationship. description	43		constructive_geometry_representation_relationship <= representation_relationship representation_relationship.description
constructive_ geometry_ association to constructive_ geometry (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => constructive_geometry_representation
constructive_ geometry_ association to geometric_model (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => shape_representation
constructive_ geometry_ association to constructive_ geometry (as constructive_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_2 -> constructive_geometry_representation
constructive_ geometry_ association to multi_language_ string (as description)	PATH		98	constructive_geometry_representation_relationship multi_language_attribute_item = \ constructive_geometry_representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
				'representation_relationship.description'}
CSG_MODEL	csg_shape_ representation	515		csg_shape_representation <= shape_representation
csg_model to b_ rep_model (as tree_root_ expression)	PATH			csg_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => shape_representation => advanced_brep_shape_representation
csg_model to boolean_result (as tree_root_ expression)	PATH			csg_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => csg_solid csg_solid.tree_root_expression -> csg_select = boolean_result
csg_model to csg_ primitive (as tree_root_ expression)	PATH			csg_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => csg_solid csg_solid.tree_root_expression -> csg_select = csg_primitive (csg_primitive = sphere sphere) (csg_primitive = block

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
				block) (csg_primitive = right_angular_wedge right_angular_wedge) (csg_primitive = torus torus) (csg_primitive = right_circular_cone right_circular_cone) (csg_primitive = right_circular_cylinder right_circular_cylinder)
csg_model to faceted_b_rep_ model (as tree_root_ expression)	PATH			csg_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => shape_representation => faceted_brep_shape_representation
csg_model to solid_replica (as tree_root_ expression)	PATH			csg_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => solid_replica
csg_model to swept_face_solid (as tree_root_ expression)	PATH			csg_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => solid_model => swept_face_solid =>

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
				(revolved_face_solid) (extruded_face_solid)
CSG_PRIMITIVE	(sphere) (block) (right_angular_wedge) (torus) (right_circular_cone) (right_circular_cylinder)	515 515 515 515 515 515		
CURVE	curve	515		
CURVE_ON_SURFACE	(surface_curve) (pcurve)	515 515		
CYLINDRICAL_SURFACE	cylindrical_surface	515		
DETAILED_ELEMENT	representation_item	515		{representation_item => (geometric_representation_item) (topological_representation_item) (mapped_item)}
name	representation_item.name	515		
detailed_element to multi_language_string (as name)	PATH		154	representation_item multi_language_attribute_item = representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
DETAILED_GEOMETRIC_MODEL_ELEMENT	geometric_representation_item	515		
DETAILED_MODEL_ELEMENT	representation_item	515		{representation_item => (geometric_representation_item) (mapped_item)}

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
EDGE_LOOP	edge_loop	515		
edge_loop to oriented_edge_ curve (as edge)	PATH			edge_loop<= path path.edge_list[i] -> oriented_edge
EDGE_TRANSITION	shape_aspect_ transition	214		shape_aspect_transition <= shape_aspect_relationship
transition	shape_aspect_ relationship.name	41		shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.name
edge_transition to oriented_edge_ curve (as edge_1)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => topological_representation_item => edge => (edge_curve) (oriented_edge)
edge_transition to oriented_edge_ curve (as edge_2)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_definition = shape_aspect

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => topological_representation_item => edge => (edge_curve) (oriented_edge)
edge_transition to vertex_point (as vertex_ joint)	PATH			shape_aspect_transition <= shape_aspect_relationship [shape_aspect_relationship.relating_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] ->] [shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] ->] representation_item => topological_representation_item => vertex => vertex_point
ELLIPSE	ellipse	515		
FACE	(face_surface) (oriented_face)	515 515		
face to face_ bound (as bound)	PATH			(face_surface <=) (oriented_face <=) face face.bounds[i] -> (face_bound) (face_bound => face_outer_bound)
face to surface (as face_ geometry)	PATH			(face_surface) (oriented_face oriented_face.face_element -> face => face_surface) face_surface.face_geometry -> surface
FACE_BOUND	(face_bound) (face_outer_bound)	515 515		
outer	face_outer_bound	515		

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
face_bound to edge_loop (as loop)	PATH			(face_bound) (face_outer_bound <= face_bound) face_bound.bound -> loop => edge_loop
face_bound to vertex_loop (as loop)	PATH			(face_bound) (face_outer_bound <= face_bound) face_bound.bound -> loop => vertex_loop
FACE_TRANSITION	shape_aspect_ transition	214		shape_aspect_transition <= shape_aspect_relationship
transition	shape_aspect_ relationship.name	41		shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.name
face_transition to oriented_edge_ curve (as edge_joint)	PATH			shape_aspect_transition <= shape_aspect_relationship [shape_aspect_relationship.relating_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] ->] [shape_aspect_relationship.related_shape_aspect -> shape_aspect shape_definition = shape_aspect

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] ->] representation_item => topological_representation_item => edge => (oriented_edge) (edge_curve)
face_transition to face (as face_1)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect -> shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => topological_representation_item => face => (oriented_face)

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
				(face_surface)
face_transition to face (as face_2)	PATH			shape_aspect_transition <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect ->
shape_aspect shape_definition = shape_aspect characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation ->
representation representation.items[i] ->
representation_item => topological_representation_item => face => (oriented_face) (face_surface)
FACETED_B_REP	(faceted_brep) (brep_with_voids)	515 515		
faceted_b_rep to faceted_closed_ shell (as outer)	PATH			(faceted_brep <=) (brep_with_voids <=) manifold_solid_brep manifold_solid_brep.outer ->
closed_shell
faceted_b_rep to faceted_closed_ shell (as void)	PATH			(faceted_brep <=) (brep_with_voids <=) manifold_solid_brep manifold_solid_brep.outer ->
closed_shell

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
FACETED_B_REP_ MODEL	faceted_brep_shape_ representation	515		faceted_brep_shape_representation <= shape_representation
faceted_b_rep_ model to faceted_ b_rep (as element)	PATH			faceted_brep_shape_representation <= shape_representation representation representation.items[i] -> representation_item => geometric_representation_item => solid_model => manifold_solid_brep => (faceted_brep) (brep_with_voids)
FACETED_CLOSED_ SHELL	closed_shell	515		
faceted_closed_ shell to faceted_ face (as shell_ boundary)	PATH			closed_shell <= connected_face_set connected_face_set.cfs_faces[i] -> face => face_surface
FACETED_FACE	face_surface	515		
faceted_face to poly_loop (as bound)	PATH			face_surface <= face face.bounds[i] -> (face_bound => face_outer_bound) (face_bound) face_bound.bound -> loop => poly_loop
faceted_face to plane (as face_ geometry)	PATH			face_surface face_surface.face_geometry -> surface => elementary_surface => plane
GEOMETRIC_MODEL	shape_representation	515	43,187	

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
description	representation. description	43		shape_representation <= representation representation.description
model_extent	uncertainty_measure_ with_unit	43		shape_representation <= representation => uncertainty_assigned_representation uncertainty_assigned_representation.uncertainty[i] -> uncertainty_measure_with_unit {uncertainty_measure_with_unit.name = 'model extent'} {uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit}
model_id	representation.id	515		shape_representation <= representation representation.id
role	representation.name	515		shape_representation <= representation representation.name {(representation.name) (representation.name = 'design shape') (representation.name = 'idealized shape')}
geometric_model to template_ instance (as additional_ elements)	PATH			shape_representation <= representation representation.items[i] -> mapped_item
geometric_model to multi_ language_string (as description)	PATH		162	shape_representation multi_language_attribute_item = shape_representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation.description'}

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_model to cartesian_ coordinate_space (as is_defined_ in)	PATH			shape_representation => representation representation.context_of_items -> representation_context => geometric_representation_context {geometric_representation_context.coordinate_space_dimension = '3'}
GEOMETRIC_MODEL_ RELATIONSHIP	representation_ relationship	43		
description	representation_ relationship. description	43		
relation_type	representation_ relationship.name	43		
geometric_model_ relationship to multi_language_ string (as description)	PATH		155	representation_relationship multi_language_attribute_item = representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
geometric_model_ relationship to external_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => externally_defined_representation
geometric_model_ relationship to geometric_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => shape_representation
geometric_model_ relationship to external_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => externally_defined_representation

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_model_ relationship to geometric_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => shape_representation
GEOMETRIC_MODEL_ VERSION	representation	515	180,33	{representation version_item = representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
version_id	identification_ assignment.assigned_ id	41	181,33	{shape_representation version_item = shape_representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
HALF_SPACE_SOLID	half_space_solid	515		
HYBRID_GEOMETRIC_ MODEL_3D	hybrid_3d_shape_ representation	214		hybrid_3d_shape_representation <= shape_representation
hybrid_geometric_ model_3d to detailed_ geometric_model_ element (as element)	PATH			hybrid_3d_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => (geometric_representation_item) (topological_representation_item)
HYPERBOLA	hyperbola	515		
LINE	line	515		

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
ORIENTED_EDGE_ CURVE	oriented_edge	515		
oriented_edge_ curve to vertex_ point (as edge_end)	PATH			oriented_edge <= edge edge.edge_end -> vertex => vertex_point
oriented_edge_ curve to b_ spline_curve (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element -> edge_curve edge_curve.edge_geometry -> curve => bounded_curve => b_spline_curve
oriented_edge_ curve to circle (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element -> edge_curve edge_curve.edge_geometry -> curve => conic => circle
oriented_edge_ curve to curve_ on_surface (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element -> edge_curve edge_curve.edge_geometry -> curve => (pcurve) (surface_curve)
oriented_edge_ curve to ellipse (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element -> edge_curve edge_curve.edge_geometry -> curve => conic => ellipse

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
oriented_edge_ curve to hyperbola (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->\ edge_curve edge_curve.edge_geometry ->\ curve =>\ conic =>\ hyperbola
oriented_edge_ curve to line (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->\ edge_curve edge_curve.edge_geometry ->\ curve =>\ line
oriented_edge_ curve to parabola (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->\ edge_curve edge_curve.edge_geometry ->\ curve =>\ conic =>\ parabola
oriented_edge_ curve to polyline (as edge_ geometry)	PATH			oriented_edge oriented_edge.edge_element ->\ edge_curve edge_curve.edge_geometry ->\ curve =>\ bounded_curve =>\ polyline
oriented_edge_ curve to vertex_ point (as edge_start)	PATH			oriented_edge <=\ edge edge.edge_start ->\ vertex =>\ vertex_point
PARABOLA	parabola	515		
PLANE	plane	515		
POINT	cartesian_point	515		

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
POLY_LOOP	poly_loop	515		
outer	face_outer_bound	515		poly_loop <= loop <- face_bound.bound face_bound => face_outer_bound
poly_loop to cartesian_point (as polygon)	PATH			poly_loop poly_loop.polygon[i] ->cartesian_point
POLYLINE	polyline	515		
RIGHT_ANGULAR_WEDGE	right_angular_wedge	515		
RIGHT_CIRCULAR_CONE	right_circular_cone	515		
RIGHT_CIRCULAR_CYLINDER	right_circular_cylinder	515		
SOLID_OF_LINEAR_EXTRUSION	extruded_face_solid	515		
SOLID_OF_REVOLUTION	revolved_face_solid	515		
SOLID_REPLICA	solid_replica	515		
solid_replica to b_rep_model (as replicated_solid)	PATH			solid_replica solid_replica.parent_solid ->solid_model => (|manifold_solid_brep|) (manifold_solid_brep =>brep_with_voids) (|manifold_solid_brep| <=) (brep_with_voids <=manifold_solid_brep <=) solid_model <=geometric_representation_item <=representation_item <-representation.items[i] representation =>shape_representation =>

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
				advanced_brep_shape_representation
solid_replica to boolean_result (as replicated_solid)	PATH			solid_replica solid_replica.parent_solid ->
solid_model => csg_solid csg_solid.tree_root_expression ->
csg_select = boolean_result boolean_result
solid_replica to csg_primitive (as replicated_solid)	PATH			solid_replica solid_replica.parent_solid ->
solid_model => csg_solid csg_solid.tree_root_expression ->
csg_select = csg_primitive (csg_primitive = sphere sphere) (csg_primitive = block block) (csg_primitive = right_angular_wedge right_angular_wedge) (csg_primitive = torus torus) (csg_primitive = right_circular_cone right_circular_cone) (csg_primitive = right_circular_cylinder right_circular_cylinder)
solid_replica to faceted_b_rep_model (as replicated_solid)	PATH			solid_replica solid_replica.parent_solid ->
solid_model => manifold_solid_brep => (faceted_brep <=) (brep_with_voids <=) manifold_solid_brep <= solid_model <= geometric_representation_item <= representation_item <-

Table 15 – Mapping table csg_model UoF (G7) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation.items[i] representation => shape_representation => faceted_brep_shape_representation
solid_replica to solid_replica (as replicated_ solid)	PATH			solid_replica solid_replica.parent_solid ->> solid_model => solid_replica
solid_replica to swept_face_solid (as replicated_ solid)	PATH			solid_replica solid_replica.parent_solid ->> solid_model => swept_face_solid => (revolved_face_solid) (extruded_face_solid)
SPHERE	sphere	515		
SPHERICAL_ SURFACE	spherical_surface	515		
SURFACE	surface	515		
SURFACE_OF_ LINEAR_EXTRUSION	surface_of_linear_ extrusion	515		
SURFACE_OF_ REVOLUTION	surface_of_ revolution	515		
SWEPT_FACE_SOLID	(revolved_face_ solid) (extruded_face_ solid)	515 515		
swept_face_solid to face (as face)	PATH			(revolved_face_solid <=) (extruded_face_solid <=) swept_face_solid swept_face_solid.swept_face ->> curve_bounded_surface <= bounded_surface <= surface <- face_surface.face_geometry face_surface

Table 15 – Mapping table csg_model UoF (G7) (concluded)

Application element	AIM element	Source	Rules	Reference path
TOROIDAL_SURFACE	toroidal_surface	515		
TORUS	torus	515		
VERTEX_LOOP	vertex_loop	515		
vertex_loop to vertex_point (as loop_vertex)	PATH			vertex_loop vertex_loop.loop_vertex ->vertex => vertex_point
VERTEX_POINT	vertex_point	515		
vertex_point to point (as vertex_ geometry)	PATH			vertex_point vertex_point.vertex_geometry ->point => cartesian_point

Table 16 – Mapping table geometrically_bounded_surface_model UoF (G8)

Application element	AIM element	Source	Rules	Reference path
B_SPLINE_CURVE	b_spline_curve	507		
B_SPLINE_SURFACE	b_spline_surface	507		
CARTESIAN_POINT	cartesian_point	507		
CIRCLE	circle	507		
COMPOSITE_CURVE	composite_curve	507		
CONICAL_SURFACE	conical_surface	507		
CONSTRUCTIVE_ GEOMETRIC_ ELEMENT	(axis2_placement_3d) (cartesian_point) (point_on_curve) (point_on_surface) (degenerate_pcurve) (curve) (surface)	507 507 507 507 507 507 507		
description	representation_item. name	507		(axis2_placement_3d <= placement <=) ((cartesian_point <=) (point_on_curve <=) (point_on_surface <=) (degenerate_pcurve <=) point <=) (surface <=) (curve <=) geometric_representation_item <= representation_item representation_item.name
constructive_ geometric_ element to axis_ placement (as element)	PATH			axis2_placement_3d <= placement
constructive_ geometric_ element to curve (as element)	IDENTICAL MAPPING			

Table 16 – Mapping table geometrically_bounded_surface_model UoF (G8) (continued)

Application element	AIM element	Source	Rules	Reference path
constructive_ geometric_ element to point (as element)	IDENTICAL MAPPING			
constructive_ geometric_ element to surface (as element)	IDENTICAL MAPPING			
CONSTRUCTIVE_ GEOMETRY	constructive_ geometry_ representation	214		constructive_geometry_representation <= representation {representation.context_of_items -> representation_context => geometric_representation_context geometric_representation_context.coordinate_space_dimension = '3'}
constructive_ geometry to constructive_ geometric_ element (as element)	PATH			constructive_geometry_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => (placement => axis2_placement_3d) (point => cartesian_point) (point => point_on_curve) (point => point_on_surface) (point => degenerate_pcurve) (curve) (surface)
CONSTRUCTIVE_ GEOMETRY_ ASSOCIATION	constructive_ geometry_ representation_ relationship	214		constructive_geometry_representation_relationship <= representation_relationship

Table 16 – Mapping table geometrically_bounded_surface_model UoF (G8) (continued)

Application element	AIM element	Source	Rules	Reference path
description	representation_ relationship. description	43		constructive_geometry_representation_relationship <= representation_relationship representation_relationship.description
constructive_ geometry_ association to constructive_ geometry (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => constructive_geometry_representation
constructive_ geometry_ association to geometric_model (as base_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => shape_representation
constructive_ geometry_ association to constructive_ geometry (as constructive_ geometry)	PATH			constructive_geometry_representation_relationship <= representation_relationship representation_relationship.rep_2 -> constructive_geometry_representation
constructive_ geometry_ association to multi_language_ string (as description)	PATH		98	constructive_geometry_representation_relationship multi_language_attribute_item = \ constructive_geometry_representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation_relationship.description'}
CURVE	curve	507		
CURVE_ON_SURFACE	(pcurve) (surface_curve)	507 507		

Table 16 – Mapping table geometrically_bounded_surface_model UoF (G8) (continued)

Application element	AIM element	Source	Rules	Reference path
CYLINDRICAL_ SURFACE	cylindrical_surface	507		
ELLIPSE	ellipse	507		
GEOMETRIC_MODEL	shape_representation	507	43,187	
description	representation. description	507		shape_representation <= representation representation.description
model_extent	uncertainty_measure_ with_unit	43		shape_representation <= representation => uncertainty_assigned_representation uncertainty_assigned_representation.uncertainty[i] -> uncertainty_measure_with_unit {uncertainty_measure_with_unit.name = 'model extent'} {uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit}
model_id	representation.id	507		shape_representation <= representation representation.id
role	representation.name	507		shape_representation <= representation representation.name {(representation.name) (representation.name = 'design shape') (representation.name = 'idealized shape')}
geometric_model to template_ instance (as additional_ elements)	PATH			shape_representation <= representation representation.items[i] -> mapped_item
geometric_model to multi_ language_string (as description)	PATH		162	shape_representation multi_language_attribute_item = shape_representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <=

Table 16 – Mapping table geometrically_bounded_surface_model UoF (G8) (continued)

Application element	AIM element	Source	Rules	Reference path
				multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation.description'}
geometric_model to cartesian_ coordinate_space (as is_defined_ in)	PATH			shape_representation => representation representation.context_of_items -> representation_context => geometric_representation_context {geometric_representation_context.coordinate_space_dimension = '3'}
GEOMETRIC_MODEL_ RELATIONSHIP	representation_ relationship	43		
description	representation_ relationship. description	43		
relation_type	representation_ relationship.name	43		
geometric_model_ relationship to multi_language_ string (as description)	PATH		155	representation_relationship multi_language_attribute_item = representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
geometric_model_ relationship to external_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => externally_defined_representation
geometric_model_ relationship to geometric_model (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation => shape_representation

Table 16 – Mapping table geometrically_bounded_surface_model UoF (G8) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_model_ relationship to external_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => externally_defined_representation
geometric_model_ relationship to geometric_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation => shape_representation
GEOMETRIC_MODEL_ VERSION	shape_representation	507	181,33	{shape_representation version_item = shape_representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
version_id	identification_ assignment.assigned_ id	41	181,33	{shape_representation version_item = shape_representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
GEOMETRIC_SET_3D	geometric_set	507		
geometric_set_3d to curve (as element)	PATH			geometric_set geometric_set.elements[i] -> curve
geometric_set_3d to point (as element)	PATH			geometric_set geometric_set.elements[i] -> point {point => (cartesian_point) (point_on_curve)

Table 16 – Mapping table geometrically_bounded_surface_model UoF (G8) (continued)

Application element	AIM element	Source	Rules	Reference path
				(point_on_surface) (degenerate_pcurve)}
geometric_set_3d to surface (as element)	PATH			geometric_set geometric_set.elements[i] -> surface
GEOMETRICALLY_ BOUNDED_SURFACE_ MODEL	geometrically_ bounded_surface_ shape_representation	507		geometrically_bounded_surface_shape_representation <= shape_representation
geometrically_ bounded_surface_ model to geometric_set_3d (as element)	PATH			geometrically_bounded_surface_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => geometric_representation_item => geometric_set
HYPERBOLA	hyperbola	507		
LINE	line	507		
OFFSET_CURVE	offset_curve_3d	507		
OFFSET_SURFACE	offset_surface	507		
PARABOLA	parabola	507		
PLANE	plane	507		
POINT	(cartesian_point) (point_on_curve) (point_on_surface) (degenerate_pcurve)	507 507 507 507		
POINT_ON_CURVE	point_on_curve	507		
POINT_ON_SURFACE	point_on_surface	507		
POLYLINE	polyline	507		
SPHERICAL_ SURFACE	spherical_surface	507		
SURFACE	surface	507		
SURFACE_OF_ LINEAR_EXTRUSION	surface_of_linear_ extrusion	507		
SURFACE_OF_ REVOLUTION	surface_of_ revolution	507		

Table 16 – Mapping table geometrically_bounded_surface_model UoF (G8) (concluded)

Application element	AIM element	Source	Rules	Reference path
TOROIDAL_SURFACE	(toroidal_surface) (degenerate_toroidal_surface)	507 507		
TRIMMED_CURVE	trimmed_curve	507		
TRIMMED_SURFACE	(rectangular_composite_surface) (rectangular_trimmed_surface) (curve_bounded_surface)	507 507 507		

Table 17 – Mapping table kinematics UoF (K1)

Application element	AIM element	Source	Rules	Reference path
ACTUATOR	pair_actuator	105		
name	pair_actuator.name	105		
actuator to kinematic_pair (as is_applied_to)	PATH			pair_actuator pair_actuator.actuated_pair -> kinematic_pair
actuator to multi_language_string (as name)	PATH		125	pair_actuator multi_language_attribute_item = pair_actuator multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
GROUND_REPRESENTATION	kinematic_ground_representation	105		
ground_representation to geometric_model (as shape_description)	PATH			kinematic_ground_representation <= representation <- representation_relationship.rep_1 representation_relationship {representation_relationship.name = 'shape description'} representation_relationship.rep_2 -> representation => shape_representation
INITIAL_CONFIGURATION	initial_state	105		
initial_configuration to pair_value (as initial_pair_values)	PATH			initial_state initial_state.pair_values[i] -> pair_value

Table 17 – Mapping table kinematics UoF (K1) (continued)

Application element	AIM element	Source	Rules	Reference path
initial_ configuration to kinematic_ mechanism (as is_applied_ to)	PATH			initial_state initial_state.applies_to_mechanism -> mechanism
INTERPOLATED_ CONFIGURATION_ SEQUENCE	interpolated_ configuration_ sequence	105		
interpolated_ configuration_ sequence to kinematic_ configuration_ interpolation (as interpolation)	PATH			interpolated_configuration_sequence interpolated_configuration_sequence.interpolation[i] -> configuration_interpolation
KINEMATIC_ CONFIGURATION_ DEFINITION	configuration_ definition	105		
kinematic_ configuration_ definition to pair_value (as pair_values)	PATH			configuration_definition configuration_definition.pair_values[i] -> pair_value
kinematic_ configuration_ definition to path_ approximation (as t_parameter)	PATH			configuration_definition configuration_definition.t_parameter -> motion_parameter_measure (motion_parameter_measure = parameter_value parameter_value) (motion_parameter_measure = measure_with_unit measure_with_unit)
KINEMATIC_ CONFIGURATION_ INTERPOLATION	configuration_ interpolation	105		

Table 17 – Mapping table kinematics UoF (K1) (continued)

Application element	AIM element	Source	Rules	Reference path
interpolation_ type	configuration_ interpolation. interpolation	105		
kinematic_ configuration_ interpolation to kinematic_ configuration_ definition (as next_ definition)	PATH			configuration_interpolation configuration_interpolation.next_configuration_definition -> configuration_definition
kinematic_ configuration_ interpolation to kinematic_ configuration_ definition (as previous_ definition)	PATH			configuration_interpolation configuration_interpolation.previous_configuration_definition -> configuration_definition
KINEMATIC_JOINT	kinematic_joint	105		
kinematic_joint to kinematic_ link (as first_link)	PATH			kinematic_joint kinematic_joint.first_link -> kinematic_link
kinematic_joint to kinematic_ pair (as pair)	PATH			kinematic_joint <- kinematic_pair.joint kinematic_pair
kinematic_joint to kinematic_ link (as second_link)	PATH			kinematic_joint kinematic_joint.second_link -> kinematic_link
KINEMATIC_LINK	kinematic_link	105		

Table 17 – Mapping table kinematics UoF (K1) (continued)

Application element	AIM element	Source	Rules	Reference path
kinematic_link to kinematic_link_representation (as representation)	PATH			kinematic_link <- kinematic_link_representation_relation.topological_aspects kinematic_link_representation_relation kinematic_link_representation_relation.geometric_aspects -> kinematic_link_representation
KINEMATIC_LINK_REPRESENTATION	kinematic_link_representation	105		
kinematic_link_representation to transformation_to_reference_frame (as additional_reference_frame)	PATH			kinematic_link_representation <= representation representation.items [i] -> representation_item => {representation_item.name = 'additional reference frame'} geometric_representation_item => (cartesian_transformation_operator => cartesian_transformation_operator_3d) (placement => axis2_placement_3d)
kinematic_link_representation to geometric_model (as physical_description)	PATH			kinematic_link_representation <= representation <- representation_relationship.rep_1 representation_relationship {representation_relationship => kinematic_link_representation_association} representation_relationship.rep_2 -> representation => shape_representation
KINEMATIC_MECHANISM	mechanism	105		
kinematic_mechanism to kinematic_link (as base)	PATH			mechanism mechanism.base -> kinematic_link

Table 17 – Mapping table kinematics UoF (K1) (continued)

Application element	AIM element	Source	Rules	Reference path
kinematic_ mechanism to transformation_ to_reference_ frame (as base_ placement)	PATH			mechanism <- representation_relationship.rep_1 representation_relationship => representation_relationship_with_transformation {representation_relationship_with_transformation => mechanism_base_placement} representation_relationship_with_transformation.transformation_operator -> transformation transformation = functionally_defined_transformation functionally_defined_transformation => cartesian_transformation_operator => cartesian_transformation_operator_3d
kinematic_ mechanism to kinematic_ structure (as structure_ definition)	PATH			mechanism mechanism.structure_definition -> kinematic_structure
KINEMATIC_PAIR	kinematic_pair	105		{kinematic_pair => (cylindrical_pair) (fully_constrained_pair) (gear_pair) (|planar_curve_pair|) (planar_curve_pair => rolling_curve_pair) (planar_curve_pair => sliding_curve_pair) (planar_pair) (point_on_planar_curve_pair) (point_on_surface_pair) (prismatic_pair) (rack_and_pinion_pair) (revolute_pair) (screw_pair) (spherical_pair) (|surface_pair|)

Table 17 – Mapping table kinematics UoF (K1) (continued)

Application element	AIM element	Source	Rules	Reference path
				(surface_pair => rolling_surface_pair) (surface_pair => sliding_surface_pair) (unconstrained_pair) (universal_pair)}
kinematic_pair to transformation_ to_reference_ frame (as transformation_ item_1)	PATH			kinematic_pair kinematic_pair.pair_placement_in_first_link_context ->rigid_placement = axis2_placement_3d axis2_placement_3d
kinematic_pair to transformation_ to_reference_ frame (as transformation_ item_2)	PATH			kinematic_pair kinematic_pair.pair_placement_in_second_link_context ->rigid_placement = axis2_placement_3d axis2_placement_3d
KINEMATIC_ STRUCTURE	kinematic_structure	105		
kinematic_ structure to kinematic_joint (as joint)	PATH			kinematic_structure kinematic_structure.joints[i] ->kinematic_joint
KINEMATIC_ TRANSFORMATION_ RESULT	kinematic_analysis_ result	105		

Table 17 – Mapping table kinematics UoF (K1) (continued)

Application element	AIM element	Source	Rules	Reference path
kinematic_ transformation_ result to kinematic_ mechanism (as is_applied_ to)	PATH			kinematic_analysis_result kinematic_analysis_result.analysed_mechanism -> mechanism
kinematic_ transformation_ result to interpolated_ configuration_ sequence (as result)	PATH			kinematic_analysis_result kinematic_analysis_result.contained_kinematic_results[i] -> kinematic_result = interpolated_configuration_sequence interpolated_configuration_sequence
PAIR_VALUE	pair_value	105		{pair_value => (cylindrical_pair_value) (gear_pair_value) (planar_pair_value) (point_on_planar_curve_pair_value) (point_on_surface_pair_value) (prismatic_pair_value) (rack_and_pinion_pair_value) (revolute_pair_value) (rolling_curve_pair_value) (rolling_surface_pair_value) (screw_pair_value) (sliding_curve_pair_value) (sliding_surface_pair_value) (spherical_pair_value) (unconstrained_pair_value) (universal_pair_value)}
pair_value to kinematic_pair (as is_applied_ to)	PATH			pair_value pair_value.applies_to_pair -> kinematic_pair

Table 17 – Mapping table kinematics UoF (K1) (continued)

Application element	AIM element	Source	Rules	Reference path
PATH_ APPROXIMATION	(parameter_value) (measure_with_unit)	41 41		
SIMPLE_PAIR_ RANGE	simple_pair_range	105		{simple_pair_range => (cylindrical_pair_range) (gear_pair_range) (planar_curve_pair_range) (planar_pair_range) (point_on_planar_curve_pair_range) (point_on_surface_pair_range) (prismatic_pair_range) (rack_and_pinion_pair_range) (revolute_pair_range) (screw_pair_range) (spherical_pair_range) (surface_pair_range) (universal_pair_range)}
simple_pair_ range to kinematic_pair (as is_applied_ to)	PATH			simple_pair_range simple_pair_range.applies_to_pair -> kinematic_pair
SIMULATION_ INFORMATION	kinematic_property_ definition	105		
simulation_ information to design_ discipline_item_ definition (as described_ object)	PATH			kinematic_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition

Table 17 – Mapping table kinematics UoF (K1) (continued)

Application element	AIM element	Source	Rules	Reference path
simulation_information to item_definition_instance_relationship (as described_object)	PATH			kinematic_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship
simulation_information to item_definition_relationship (as described_object)	PATH			kinematic_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship
simulation_information to item_instance (as described_object)	PATH			kinematic_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition (characterized_product_definition = product_definition product_definition => product_definition_occurrence) (characterized_product_definition = product_definition_relationship product_definition_relationship => product_definition_usage => assembly_component_usage)
simulation_information to item_instance_relationship (as described_object)	PATH			kinematic_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}

Table 17 – Mapping table kinematics UoF (K1) (continued)

Application element	AIM element	Source	Rules	Reference path
simulation_information to item_solution (as described_object)	PATH			kinematic_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition => solution_based_product_definition
simulation_information to product_class (as described_object)	PATH			kinematic_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_object characterized_object => product_class
simulation_information to product_component (as described_object)	PATH			kinematic_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_object characterized_object => product_component
simulation_information to product_function (as described_object)	PATH			kinematic_property_definition <= property_definition property_definition.definition -> characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition {product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <=

Table 17 – Mapping table kinematics UoF (K1) (continued)

Application element	AIM element	Source	Rules	Reference path
				application_context_element application_context_element.name = 'functionality']}
simulation_ information to ground_ representation (as ground)	PATH			kinematic_property_definition kinematic_property_definition.ground_definition ->
characterized_definition (characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition) (characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship characterized_product_definition = product_definition_relationship characterized_definition = characterized_product_definition) (characterized_definition = characterized_object characterized_object characterized_definition = characterized_object) characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->
representation => kinematic_ground_representation

Table 17 – Mapping table kinematics UoF (K1) (concluded)

Application element	AIM element	Source	Rules	Reference path
simulation_ information to kinematic_ mechanism (as mechanism)	PATH			kinematic_property_definition <- mechanism.containing_property mechanism
TOOL_ATTACHMENT_ POINT_FRAME	axis2_placement_3d	42		{axis2_placement_3d <= geometric_representation_item <= representation_item <- {representation_item.name = 'tool attachment point frame'} representation.items[i] representation => kinematic_link_representation}
TOOL_CENTER_ POINT_FRAME	axis2_placement_3d	42		{axis2_placement_3d <= geometric_representation_item <= representation_item <- {representation_item.name = 'tool center point frame'} representation.items[i] representation => kinematic_link_representation}
TRANSFORMATION_ TO_REFERENCE_ FRAME	(cartesian_ transformation_ operator_3d) (axis2_placement_3d)	42 42		

Table 18 – Mapping table measured_data UoF (MD1)

Application element	AIM element	Source	Rules	Reference path
POINT_DIRECTION	axis1_placement	42		
vector	axis1_placement.axis	42		
point_direction to cartesian_point (as location)	PATH			axis1_placement <= placement placement.location -> cartesian_point
POINT_DIRECTION_MODEL	point_placement_shape_representation	214		point_placement_shape_representation <= shape_representation
point_direction_model to point_sequence (as point)	PATH			point_placement_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => compound_representation_item {compound_representation_item <= representation_item representation_item.name = 'point sequence'}
point_direction_model to point_direction_sequence (as point_direction)	PATH			point_placement_shape_representation <= shape_representation <= representation representation.items[i] -> representation_item => compound_representation_item {compound_representation_item <= representation_item representation_item.name = 'point placement sequence'}
POINT_DIRECTION_SEQUENCE	compound_representation_item	43		{compound_representation_item <= representation_item representation_item.name = 'point placement sequence'}
point_direction_sequence to point_direction (as element)	PATH			compound_representation_item compound_representation_item.item_element -> compound_item_definition compound_item_definition = list_representation_item list_representation_item[i]: representation_item representation_item =>

Table 18 – Mapping table measured_data UoF (MD1) (concluded)

Application element	AIM element	Source	Rules	Reference path
				geometric_representation_item => placement => axis1_placement
point_direction_ sequence to point_on_surface (as element)	PATH			compound_representation_item compound_representation_item.item_element ->> compound_item_definition compound_item_definition = list_representation_item list_representation_item[i]: representation_item representation_item => geometric_representation_item => point => point_on_surface
POINT_SEQUENCE	compound_ representation_item	43		{compound_representation_item <= representation_item representation_item.name = 'point sequence'}
point_sequence to cartesian_ point (as element)	PATH			compound_representation_item compound_representation_item.item_element ->> compound_item_definition compound_item_definition = list_representation_item list_representation_item[i]: representation_item representation_item => geometric_representation_item => point => cartesian_point

Table 19 – Mapping table geometric_presentation UoF (P1)

Application element	AIM element	Source	Rules	Reference path
APPEARANCE_ ASSIGNMENT	presentation_style_ assignment	504		
appearance_ assignment to curve_appearance (as component)	PATH			presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select = curve_style curve_style
appearance_ assignment to hatching (as component)	PATH		30	presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select = fill_area_style fill_area_style.fill_styles[i] -> fill_style_select (fill_style_select = fill_area_style_hatching fill_area_style_hatching) (fill_style_select = externally_defined_hatch_style externally_defined_hatch_style)
appearance_ assignment to point_appearance (as component)	PATH			presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select = point_style point_style
appearance_ assignment to solid_fill_area (as component)	PATH		30	presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select = fill_area_style fill_area_style.fill_styles[i] -> fill_style_select fill_style_select = fill_area_style_colour fill_area_style_colour
appearance_ assignment to surface_ appearance (as component)	PATH			presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select = surface_style_usage surface_style_usage
appearance_ assignment to text_appearance (as component)	PATH			presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select = text_style text_style => (text_style_with_box_characteristics)

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
				([text_style_with_mirror] [text_style_with_box_characteristics])
appearance_ assignment to tiling (as component)	PATH		30	presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select = fill_area_style fill_area_style.fill_styles[i] -> fill_style_select (fill_style_select = fill_area_style_tiles fill_area_style_tiles) (fill_style_select = externally_defined_tile_style externally_defined_tile_style)
appearance_ assignment to vector_ appearance (as component)	PATH			presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select = curve_style curve_style => vector_style
CONTEXT_ DEPENDENT_ APPEARANCE_ ASSIGNMENT	presentation_style_ by_context	517	42	
context_ dependent_ appearance_ assignment to drawing_sheet (as appearance_ context)	PATH			presentation_style_by_context presentation_style_by_context.style_context -> style_context_select style_context_select = representation representation => presentation_representation => presentation_area => drawing_sheet_revision
context_ dependent_ appearance_ assignment to group (as appearance_ context)	PATH			presentation_style_by_context presentation_style_by_context.style_context -> style_context_select style_context_select = representation_item representation_item => rep_item_group

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
context_ dependent_ appearance_ assignment to layer (as appearance_ context)	PATH			presentation_style_by_context presentation_style_by_context.style_context -> style_context_select style_context_select = representation representation => presentation_representation <- presentation_layer_usage.presentation presentation_layer_usage presentation_layer_usage.assignment -> presentation_layer_assignment
context_ dependent_ appearance_ assignment to model_image (as appearance_ context)	PATH			presentation_style_by_context presentation_style_by_context.style_context -> style_context_select style_context_select = representation_item representation_item => mapped_item => camera_image => (camera_image_2d_with_scale) (camera_image_3d_with_scale)
context_ dependent_ appearance_ assignment to view (as appearance_ context)	PATH			presentation_style_by_context presentation_style_by_context.style_context -> style_context_select style_context_select = representation representation => presentation_representation => presentation_view
context_ dependent_ appearance_ assignment to view_area (as appearance_ context)	PATH			presentation_style_by_context presentation_style_by_context.style_context -> style_context_select style_context_select = representation representation => presentation_representation => (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
CURVE_APPEARANCE	curve_style	504, 517	20	
colour	curve_style.curve_colour	517	194,40, 18	{curve_style curve_style.curve_colour -> colour => (colour_specification => colour_rgb) (pre_defined_colour => draughting_pre_defined_colour)}
font	curve_style.curve_font	517	10	{curve_style curve_style.curve_font -> curve_font_or_scaled_curve_font_select curve_font_or_scaled_curve_font_select = curve_style_font_select curve_style_font_select (curve_style_font_select = curve_style_font curve_style_font) (curve_style_font_select = pre_defined_curve_font pre_defined_curve_font => draughting_pre_defined_curve_font) (curve_style_font_select = externally_defined_curve_font externally_defined_curve_font)}
width	curve_style.curve_width	517		{curve_style curve_style.curve_width -> size_select (size_select = measure_with_unit measure_with_unit => length_measure_with_unit) (size_select = positive_length_measure)}
MODEL_IMAGE	(camera_image_2d_with_scale) (camera_image_3d_with_scale)	214 517	190	(camera_image_2d_with_scale <=) (camera_image_3d_with_scale <=) camera_image
clipping	(camera_model_d2.view_window) (view_volume.view_window)	46 517	191	(camera_image_2d_with_scale <=) (camera_image_3d_with_scale <=) camera_image <= mapped_item

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
				mapped_item.mapping_source -> representation_map {representation_map => camera_usage} representation_map.mapping_origin -> representation_item => geometric_representation_item => camera_model => (camera_model_d2 camera_model_d2.view_window) (camera_model_d3 camera_model_d3.perspective_of_volume -> view_volume view_volume.view_window)
name	representation_item. name	517		(camera_image_2d_with_scale <=) (camera_image_3d_with_scale <=) camera_image <= mapped_item <= representation_item representation_item.name
scale	(camera_image_2d_ with_scale.scale) (camera_image_3d_ with_scale.scale)	214 517		
view_projection	view_volume. projection_type	517		(camera_image_2d_with_scale <=) (camera_image_3d_with_scale <=) camera_image <= mapped_item mapped_item.mapping_source -> representation_map {representation_map => camera_usage} representation_map.mapping_origin -> representation_item => geometric_representation_item => camera_model =>

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
				camera_model_d3 {(camera_model_d3) (camera_model_d3 => camera_model_d3_with_hlhsr)} camera_model_d3.perspective_of_volume -> view_volume view_volume.projection_type
view_volume_ placement	(camera_model_d3. view_reference_ system) (planar_box. placement)	517 46		(camera_image_2d_with_scale <=) (camera_image_3d_with_scale <=) camera_image <= mapped_item mapped_item.mapping_source -> representation_map {representation_map => camera_usage} representation_map.mapping_origin -> representation_item => geometric_representation_item => camera_model => (camera_model_d3 camera_model_d3.view_reference_system) (camera_model_d2 camera_model_d2.view_window -> planar_box planar_box.placement)
model_image to external_model (as is_ displaying)	PATH			(camera_image_2d_with_scale <=) (camera_image_3d_with_scale <=) camera_image <= mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => externally_defined_representation

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
model_image to styled_model (as is_displaying)	PATH			(camera_image_2d_with_scale <=) (camera_image_3d_with_scale <=) camera_image <= mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => (draughting_model) (mechanical_design_geometric_presentation_representation)
model_image to view (as is_presented_in)	PATH			(camera_image_2d_with_scale <=) (camera_image_3d_with_scale <=) camera_image <= mapped_item <= representation_item <- representation.items[i] representation => presentation_representation => presentation_view
model_image to multi_language_string (as name)	PATH		90,91	(camera_image_2d_with_scale multi_language_attribute_item = camera_image_2d_with_scale) (camera_image_3d_with_scale multi_language_attribute_item = camera_image_3d_with_scale) multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation_item.name'}
POINT_APPEARANCE	point_style	517		
colour	point_style.marker_colour	517	194,40, 18	{point_style point_style.marker_colour -> colour => (colour_specification => colour_rgb)

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
				(pre_defined_colour => draughting_pre_defined_colour)}
marker	point_style.marker	517		{point_style.marker -> marker_select marker_select = pre_defined_marker pre_defined_marker => pre_defined_point_marker_symbol}
STYLED_ELEMENT	styled_item	517		
invisible	invisibility. invisible_items	517		styled_item invisible_item = styled_item invisible_item <- invisibility.invisible_items[i]
styled_element to appearance_ assignment (as appearance)	PATH			styled_item styled_item.styles[i] -> presentation_style_assignment
STYLED_GEOMETRIC_ MODEL	(draughting_model) (mechanical_design_ geometric_ presentation_ representation)	214 517		(draughting_model <=) (mechanical_design_geometric_presentation_representation <=) representation
styled_geometric_ model to geometric_model (as styled_ geometry)	PATH			(draughting_model <=) (mechanical_design_geometric_presentation_representation <=) representation representation.items[i] -> representation_item => mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => shape_representation
STYLED_MODEL	(draughting_model) (mechanical_design_ geometric_ presentation_	214 517	43,71	(draughting_model <=) (mechanical_design_geometric_presentation_representation <=) representation

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
	representation)			
description	representation. description	43		(draughting_model <=) (mechanical_design_geometric_presentation_representation <=) representation representation.description
model_id	representation.id	43		(draughting_model <=) (mechanical_design_geometric_presentation_representation <=) representation representation.id
styled_model to template_ instance (as additional_ elements)	PATH			(draughting_model <= representation representation.items[i] -> representation_item => mapped_item {mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => draughting_model}) (mechanical_design_geometric_presentation_representation <= representation representation.items[i] -> representation_item => mapped_item {mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => mechanical_design_geometric_presentation_representation})
styled_model to multi_language_ string (as description)	PATH		106,121	(draughting_model multi_language_attribute_item = draughting_model) (mechanical_design_geometric_presentation_representation multi_language_attribute_item = \ mechanical_design_geometric_presentation_representation) multi_language_attribute_item <-

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation.description'}
styled_model to styled_element (as element)	PATH		51	(draughting_model <=) (mechanical_design_geometric_presentation_representation <=) representation representation.items[i] -> representation_item => styled_item
STYLED_MODEL_ ELEMENT	styled_item	517		
styled_model_ element to detailed_model_ element (as element)	PATH		189	styled_item styled_item.item -> representation_item
STYLED_MODEL_ RELATIONSHIP	representation_ relationship	43		
description	representation_ relationship. description	43		
relation_type	representation_ relationship.name	43		
styled_model_ relationship to multi_language_ string (as description)	PATH		155	representation_relationship multi_language_attribute_item = representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
styled_model_ relationship to styled_model (as related)	PATH			representation_relationship representation_relationship.rep_2 representation => (draughting_model) (mechanical_design_geometric_presentation_representation)
styled_model_ relationship to styled_model (as relating)	PATH			representation_relationship representation_relationship.rep_1 representation => (draughting_model) (mechanical_design_geometric_presentation_representation)
STYLED_MODEL_ VERSION	(draughting_model) (mechanical_design_ geometric_ presentation_ representation)	214 517	171,172, 33	{(draughting_model version_item = draughting_model) (mechanical_design_geometric_presentation_representation version_item = \ mechanical_design_geometric_presentation_representation) version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id'}
version_id	identification_ assignment.assigned_ id	41	171,172, 33	(draughting_model version_item = draughting_model) (mechanical_design_geometric_presentation_representation version_item = \ mechanical_design_geometric_presentation_representation) version_item <- version_assignment.items[1] version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
SURFACE_ APPEARANCE	surface_style_usage	517		
side	surface_style_usage. side	517		
surface_ appearance to surface_ appearance_ element (as element)	PATH			surface_style_usage surface_style_usage.style -> surface_side_style_select surface_side_style_select = surface_side_style surface_side_style.styles[i] -> surface_style_element_select (surface_style_element_select = surface_style_fill_area surface_style_fill_area) (surface_style_element_select = surface_style_boundary surface_style_boundary) (surface_style_element_select = surface_style_parameter_line surface_style_parameter_line)
SURFACE_ APPEARANCE_ ELEMENT	(surface_style_ boundary) (surface_style_ parameter_line) (surface_style_fill_ area)	517 517 46	30	{(surface_style_parameter_line surface_style_parameter_line.direction_counts[i] -> direction_count_select (direction_count_select = u_direction_count u_direction_count) (direction_count_select = v_direction_count v_direction_count)) (surface_style_fill_area surface_style_fill_area.fill_area -> fill_area_style fill_area_style.fill_styles[i] -> fill_style_select fill_style_select = fill_area_style_colour fill_area_style_colour)}

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
surface_ appearance_ element to visual_ appearance (as surface_ condition)	PATH			(surface_style_fill_area surface_style_element_select = surface_style_fill_area) (surface_style_boundary surface_style_element_select = surface_style_boundary) (surface_style_parameter_line surface_style_element_select = surface_style_parameter_line) surface_style_element_select <- surface_side_style.styles[i] surface_side_style surface_side_style_select = surface_side_style surface_side_style_select <- surface_style_usage.style surface_style_usage presentation_style_select = surface_style_usage presentation_style_select <- presentation_style_assignment.styles[i] presentation_style_assignment <- styled_item.styles[i] styled_item styled_item.item -> representation_item <- representation.items[i] representation {[representation representation.description = 'visual appearance'] [representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition => surface_condition]}

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
VECTOR_ APPEARANCE	vector_style	214	41,20	vector_style <= [pre_defined_terminator_symbol <= pre_defined_symbol] [curve_style]
arrow_head_type	pre_defined_ terminator_symbol	214		vector_style <= pre_defined_terminator_symbol <= pre_defined_symbol
arrow_line_font	curve_style.curve_ font	517	195,10	vector_style <= curve_style curve_style.curve_font -> curve_font_or_scaled_curve_font_select curve_font_or_scaled_curve_font_select = curve_style_font_select curve_style_font_select (curve_style_font_select = curve_style_font curve_style_font) (curve_style_font_select = pre_defined_curve_font pre_defined_curve_font => draughting_pre_defined_curve_font) (curve_style_font_select = externally_defined_curve_font externally_defined_curve_font)
VIEW	presentation_view	517		{presentation_view <= (presentation_representation) (presentation_representation => mechanical_design_geometric_presentation_representation)}
description	representation. description	43		presentation_view <= (presentation_representation <=) (presentation_representation <= {presentation_representation => mechanical_design_geometric_presentation_representation}) representation representation.description
id	representation.id	43		presentation_view <= (presentation_representation <=)

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
				(presentation_representation <= {presentation_representation => mechanical_design_geometric_presentation_representation}) representation representation.id
name	representation.name	43		presentation_view <= (presentation_representation <=) (presentation_representation <= {presentation_representation => mechanical_design_geometric_presentation_representation}) representation representation.name
view to multi_ language_string (as description)	PATH		128	presentation_view multi_language_attribute_item = presentation_view multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation.description'}
view to drawing_ sheet (as layout)	PATH		54	presentation_view <= (presentation_representation <=) (presentation_representation <= {presentation_representation => mechanical_design_geometric_presentation_representation}) representation <- representation_map.mapped_representation representation_map <- mapped_item.mapping_source mapped_item <=

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item <- representation.items[i] representation => presentation_representation => presentation_area => drawing_sheet_revision
view to view_ area (as layout)	PATH			presentation_view <= (presentation_representation <=) (presentation_representation <= {presentation_representation => mechanical_design_geometric_presentation_representation}) representation <- representation_map.mapped_representation representation_map <- mapped_item.mapping_source mapped_item <= representation_item <- representation.items[i] representation => presentation_representation => (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)
view to multi_ language_string (as name)	PATH		128	presentation_view multi_language_attribute_item = presentation_view multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'representation.name'}

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
view to transformation_ 2d (as transformation)	PATH			presentation_view <= presentation_representation <= representation <- representation_map.mapped_representation representation_map [representation_map.mapping_origin {representation_map.mapping_origin -> representation_item => geometric_representation_item => placement => axis2_placement_2d}] [representation_map <- mapped_item.mapping_source mapped_item {mapped_item <- representation.items[i] representation => presentation_representation => presentation_area (presentation_area) (presentation_area => drawing_sheet_revision)} mapped_item.mapping_target {mapped_item.mapping_target -> representation_item => geometric_representation_item => (placement => axis2_placement_2d) (cartesian_transformation_operator => cartesian_transformation_operator_2d)}]

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
view to cartesian_ coordinate_space_ 2d (as view_space)	PATH			presentation_view <= (presentation_representation <=) (presentation_representation <= {presentation_representation => mechanical_design_geometric_presentation_representation}) representation representation.context_of_items -> representation_context => geometric_representation_context {geometric_representation_context.coordinate_space_dimension = 2}
VIEW_AREA	(presentation_area) (mechanical_design_ geometric_ presentation_area)	46 517		mechanical_design_geometric_presentation_area <= presentation_area
description	representation. description	43		(presentation_area <=) (mechanical_design_geometric_presentation_area <= presentation_area <=) presentation_representation <= representation representation.description
id	representation.id	43		(presentation_area <=) (mechanical_design_geometric_presentation_area <= presentation_area <=) presentation_representation <= representation representation.id
name	representation.name	43		(presentation_area <=) (mechanical_design_geometric_presentation_area <= presentation_area <=) presentation_representation <= representation representation.name
revision_id	identification_ assignment.assigned_	41	174,33	presentation_area version_item = presentation_area

Table 19 – Mapping table geometric_presentation UoF (P1) (continued)

Application element	AIM element	Source	Rules	Reference path
	id			version_item <- version_assignment.items[1] version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
size	presentation_size. size	46		(presentation_area) (mechanical_design_geometric_presentation_area <= presentation_area) presentation_size_assignment_select = presentation_area presentation_size_assignment_select <- presentation_size.unit presentation_size presentation_size.size
view_area to cartesian_ coordinate_space_ 2d (as area_space)	PATH			(presentation_area <=) (mechanical_design_geometric_presentation_area <= presentation_area <=) presentation_representation <= representation representation.context_of_items -> representation_context => geometric_representation_context {geometric_representation_context.coordinate_space_dimension = 2}
view_area to multi_language_ string (as description)	PATH		126	(presentation_area) (mechanical_design_geometric_presentation_area <= presentation_area) multi_language_attribute_item = presentation_area multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation.description'}

Table 19 – Mapping table geometric_presentation UoF (P1) (concluded)

Application element	AIM element	Source	Rules	Reference path
view_area to multi_language_ string (as name)	PATH		126	(presentation_area) (mechanical_design_geometric_presentation_area <= presentation_area) multi_language_attribute_item = presentation_area multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation.name'}

Table 20 – Mapping table annotated_presentation UoF (P2)

Application element	AIM element	Source	Rules	Reference path
ANNOTATION_CURVE	annotation_curve_ occurrence	504		
annotation_curve_ type	curve_style.name	46		annotation_curve_occurrence <= annotation_occurrence <= {annotation_occurrence => draughting_annotation_occurrence} styled_item styled_item.styles[i] -> presentation_style_assignment presentation_style_assignment.styles[i] -> presentation_style_select presentation_style_select = curve_style curve_style curve_style.name
ANNOTATION_ ELEMENT	annotation_ occurrence	504		
ANNOTATION_MODEL	draughting_model	214		draughting_model <= representation {representation.items[i] -> representation_item => (styled_item => annotation_occurrence => draughting_annotation_occurrence) (geometric_representation_item => draughting_callout => draughting_elements)}
annotation_model to cartesian_ coordinate_space (as is_defined_ in)	PATH			draughting_model <= representation representation.context_of_items -> representation_context => geometric_representation_context
ANNOTATION_ SUBFIGURE	annotation_subfigure_ occurrence	504		annotation_subfigure_occurrence <= annotation_symbol_occurrence <= {annotation_occurrence => draughting_annotation_occurrence}

Table 20 – Mapping table annotated_presentation UoF (P2) (continued)

Application element	AIM element	Source	Rules	Reference path
ANNOTATION_ SYMBOL	(annotation_symbol_ occurrence) (draughting_ elements)	504 506	45,29	draughting_elements <= draughting_callout
AREA_PLACED_ ANNOTATION	(draughting_ annotation_ occurrence) (draughting_ elements)	504 506		(draughting_annotation_occurrence <= annotation_occurrence) (draughting_elements <= draughting_callout)
invisible	invisibility. invisible_items	517		(draughting_annotation_occurrence <= annotation_occurrence <=) (draughting_elements <= draughting_callout draughting_callout.contents[i] -> draughting_callout_element (draughting_callout_element = annotation_curve_occurrence annotation_curve_occurrence <=) (draughting_callout_element = annotation_symbol_occurrence annotation_symbol_occurrence <=) (draughting_callout_element = annotation_text_occurrence annotation_text_occurrence <=) annotation_occurrence <=) styled_item invisible_item = styled_item invisible_item <- invisibility.invisible_items[i]
area_placed_ annotation to appearance_ assignment (as appearance)	PATH			(draughting_annotation_occurrence <= annotation_occurrence <=) (draughting_elements <= draughting_callout <= geometric_representation_item <= representation_item <- styled_item.item)

Table 20 – Mapping table annotated_presentation UoF (P2) (continued)

Application element	AIM element	Source	Rules	Reference path
				styled_item styled_item.styles[i] -> presentation_style_assignment
area_placed_ annotation to annotation_ element (as element)	IDENTICAL MAPPING			{draughting_annotation_occurrence <= annotation_occurrence}
area_placed_ annotation to drawing_sheet (as reference_ area)	PATH		53	(draughting_annotation_occurrence <= annotation_occurrence <= styled_item <=) (draughting_elements <= draughting_callout <= geometric_representation_item <=) representation_item <- representation.items[i] representation <- representation_map.mapped_representation representation_map <- mapped_item.mapping_source mapped_item <= representation_item <- representation.items[i] representation => presentation_representation => presentation_area => drawing_sheet_revision
area_placed_ annotation to view_area (as reference_ area)	PATH			(draughting_annotation_occurrence <= annotation_occurrence <= styled_item <=) (draughting_elements <= draughting_callout <=

Table 20 – Mapping table annotated_presentation UoF (P2) (continued)

Application element	AIM element	Source	Rules	Reference path
				geometric_representation_item <=) representation_item <- representation.items[i] representation <- representation_map.mapped_representation representation_map <- mapped_item.mapping_source mapped_item <= representation_item <- representation.items[i] representation => presentation_representation => presentation_area
DETAILED_ELEMENT	representation_item	43		{representation_item => (styled_item) (mapped_item)}
name	representation_item. name	43		
detailed_element to multi_ language_string (as name)	PATH		154	representation_item multi_language_attribute_item = representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
DRAUGHTING_ CALLOUT	draughting_elements	506		draughting_elements <= draughting_callout {draughting_callout => (datum_feature_callout) (datum_target_callout) (geometrical_tolerance_callout)}
FILL_AREA	annotation_fill_area_ occurrence	504		

Table 20 – Mapping table annotated_presentation UoF (P2) (continued)

Application element	AIM element	Source	Rules	Reference path
reference_point	annotation_fill_area_occurrence.fill_style_target	504		
fill_area to annotation_curve (as boundary)	PATH			annotation_fill_area_occurrence <= annotation_occurrence <= {annotation_occurrence => draughting_annotation_occurrence} styled_item styled_item.item -> representation_item => geometric_representation_item => annotation_fill_area annotation_fill_area.boundaries[i] -> curve <= geometric_representation_item <= representation_item <- styled_item.item styled_item => annotation_occurrence => {annotation_occurrence => draughting_annotation_occurrence} annotation_curve_occurrence
HATCHING	(fill_area_style_hatching) (externally_defined_hatch_style)	504 504		
LEADER	leader_curve	506		
leader to annotation_symbol (as terminator)	PATH		199,41	leader_curve <= annotation_curve_occurrence <- terminator_symbol.annotated_curve terminator_symbol <= {terminator_symbol => leader_terminator} annotation_symbol_occurrence {annotation_symbol_occurrence <= annotation_occurrence <=

Table 20 – Mapping table annotated_presentation UoF (P2) (continued)

Application element	AIM element	Source	Rules	Reference path
				{annotation_occurrence => draughting_annotation_occurrence} styled_item styled_item.item ->* representation_item => geometric_representation_item => defined_symbol defined_symbol.definition ->* defined_symbol_select defined_symbol_select = pre_defined_symbol pre_defined_symbol => pre_defined_terminator_symbol}
SOLID_FILL_AREA	fill_area_style_ colour	504	31	
TEXT	annotation_text_ occurrence	504	196,200	{annotation_text_occurrence <= annotation_occurrence <= {annotation_occurrence => draughting_annotation_occurrence} styled_item styled_item.item ->* representation_item => (geometric_representation_item => text_literal {text_literal.font ->* font_select (font_select = pre_defined_text_font pre_defined_text_font => draughting_pre_defined_text_font) (font_select = externally_defined_text_font externally_defined_text_font)} (text_literal) (text_literal => text_literal_with_extent) (text_literal => text_literal_with_delineation =>

Table 20 – Mapping table annotated_presentation UoF (P2) (continued)

Application element	AIM element	Source	Rules	Reference path
				draughting_text_literal_with_delineation) (text_literal => text_literal_with_blanking_box) (text_literal => text_literal_with_associated_curves)) (geometric_representation_item => (composite_text) (composite_text => composite_text_with_extent) (composite_text => composite_text_with_blanking_box) (composite_text => composite_text_with_associated_curves)) (mapped_item => {mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => text_string_representation} annotation_text)}
TEXT_APPEARANCE	(text_style_with_box_characteristics) ([text_style_with_mirror] [text_style_with_box_characteristics])	504 504 504	47,46	
TILING	(fill_area_style_tiles) (externally_defined_tile_style)	504 504	59	
VIEW_AREA_RELATIONSHIP	area_presentation_relationship	214		area_presentation_relationship <= representation_relationship
description	representation_relationship.	43		area_presentation_relationship <= representation_relationship

Table 20 – Mapping table annotated_presentation UoF (P2) (continued)

Application element	AIM element	Source	Rules	Reference path
	description			representation_relationship.description
relation_type	representation_ relationship.name	43		area_presentation_relationship <= representation_relationship representation_relationship.name {(representation_relationship.name) (representation_relationship.name = 'copy') (representation_relationship.name = 'derivation') (representation_relationship.name = 'hierarchy') (representation_relationship.name = 'substitution')}
view_area_ relationship to multi_language_ string (as description)	PATH			area_presentation_relationship multi_language_attribute_item = \ area_presentation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation_relationship.description'}
view_area_ relationship to view_area (as related)	PATH			area_presentation_relationship <= representation_relationship representation_relationship.rep_2 -> representation => presentation_representation => (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)
view_area_ relationship to view_area (as relating)	PATH			area_presentation_relationship <= representation_relationship representation_relationship.rep_1 -> representation => presentation_representation => (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)

Table 20 – Mapping table annotated_presentation UoF (P2) (continued)

Application element	AIM element	Source	Rules	Reference path
VIEW_PLACED_ ANNOTATION	(draughting_ annotation_ occurrence) (draughting_ elements)	504 506		(draughting_annotation_occurrence <= annotation_occurrence) (draughting_elements <= draughting_callout)
invisible	invisibility. invisible_items			(draughting_annotation_occurrence <= annotation_occurrence <=) (draughting_elements <= draughting_callout draughting_callout.contents[i] -> draughting_callout_element (draughting_callout_element = annotation_curve_occurrence annotation_curve_occurrence <=) (draughting_callout_element = annotation_symbol_occurrence annotation_symbol_occurrence <=) (draughting_callout_element = annotation_text_occurrence annotation_text_occurrence <=) annotation_occurrence <=) styled_item invisible_item = styled_item invisible_item <- invisibility.invisible_items[i]
view_placed_ annotation to appearance_ assignment (as appearance)	PATH			(draughting_annotation_occurrence <= annotation_occurrence <=) (draughting_elements <= draughting_callout <= geometric_representation_item <= representation_item <- styled_item.item) styled_item styled_item.styles[i] -> presentation_style_assignment

Table 20 – Mapping table annotated_presentation UoF (P2) (continued)

Application element	AIM element	Source	Rules	Reference path
view_placed_ annotation to annotation_ element (as element)	IDENTICAL MAPPING		55	{draughting_annotation_occurrence <= annotation_occurrence}
view_placed_ annotation to view (as reference_ view)	PATH			(draughting_annotation_occurrence <= annotation_occurrence <= styled_item <=) (draughting_elements <= draughting_callout <= geometric_representation_item <=) representation_item <- representation.items[i] representation <- representation_map.mapped_representation representation_map <- mapped_item.mapping_source mapped_item <= representation_item <- representation.items[i] representation => presentation_representation => presentation_view
VIEW_ RELATIONSHIP	presentation_view_ relationship	214		presentation_view_relationship <= representation_relationship
description	representation_ relationship. description	43		presentation_view_relationship <= representation_relationship representation_relationship.description
relation_type	representation_ relationship.name	43		presentation_view_relationship <= representation_relationship representation_relationship.name {(representation_relationship.name) (representation_relationship.name = 'dependency')}

Table 20 – Mapping table annotated_presentation UoF (P2) (concluded)

Application element	AIM element	Source	Rules	Reference path
view_ relationship to multi_language_ string (as description)	PATH		129	presentation_view_relationship multi_language_attribute_item = \ presentation_view_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation_relationship.description'}
view_ relationship to view (as related)	PATH			presentation_view_relationship <= representation_relationship representation_relationship.rep_2 -> representation => presentation_representation => presentation_view
view_ relationship to view (as relating)	PATH			presentation_view_relationship <= representation_relationship representation_relationship.rep_1 -> representation => presentation_representation => presentation_view

Table 21 – Mapping table shaded_presentation UoF (P3)

Application element	AIM element	Source	Rules	Reference path
CURVE_APPEARANCE	(curve_style) (curve_style_rendering)	517 46	20	
colour	curve_style.curve_colour	517	194,40,18	{curve_style curve_style.curve_colour -> colour => (colour_specification => colour_rgb) (pre_defined_colour => draughting_pre_defined_colour)}
font	curve_style.curve_font	517	195	{curve_style curve_style.curve_font -> curve_font_or_scaled_curve_font_select curve_font_or_scaled_curve_font_select = curve_style_font_select curve_style_font_select (curve_style_font_select = curve_style_font curve_style_font curve_style_font.pattern_list[i] -> curve_style_font_pattern) (curve_style_font_select = pre_defined_curve_font pre_defined_curve_font => draughting_pre_defined_curve_font) (curve_style_font_select = externally_defined_curve_font externally_defined_curve_font)}
width	curve_style.curve_width	517		{curve_style curve_style.curve_width -> size_select (size_select = measure_with_unit measure_with_unit => length_measure_with_unit) (size_select = positive_length_measure)}
HIDDEN_ELEMENT_HANDLING	(camera_model_d3_with_hlhsr) (hidden_element_over_riding_styled_item)	46 214		(camera_model_d3_with_hlhsr) (hidden_element_over_riding_styled_item <= context_dependent_over_riding_styled_item)

Table 21 – Mapping table shaded_presentation UoF (P3) (continued)

Application element	AIM element	Source	Rules	Reference path
switch	camera_model_d3_with_hlhsr.hidden_line_surface_removal	46		
hidden_element_handling to appearance_assignment (as hidden_element_appearance)	PATH			hidden_element_over_riding_styled_item <= context_dependent_over_riding_styled_item <= over_riding_styled_item <= styled_item styled_item.styles[i] -> presentation_style_assignment
hidden_element_handling to model_image (as is_defined_for)	PATH			(camera_model_d3_with_hlhsr <= camera_model_d3 <= camera_model <= geometric_representation_item <= representation_item <- representation_map.mapping_origin representation_map <- {representation_map => camera_usage} mapped_item.mapping_source) (hidden_element_over_riding_styled_item <= context_dependent_over_riding_styled_item <= over_riding_styled_item <= styled_item styled_item.item -> representation_item =>) mapped_item => camera_image => (camera_image_2d_with_scale) (camera_image_3d_with_scale)
LIGHT_SOURCE	light_source	46		
colour	light_source.light_colour	46	194,40, 18	{colour => (colour_rgb) (pre_defined_colour =>

Table 21 – Mapping table shaded_presentation UoF (P3) (continued)

Application element	AIM element	Source	Rules	Reference path
				draughting_pre_defined_colour)}
light_type	(light_source_ambient) (light_source_directional) (light_source_spot) (light_source_positional)	46 46 46 46		light_source => (light_source_ambient) (light_source_directional) (light_source_spot) (light_source_positional)
light_source to model_image (as view)	PATH			light_source <= geometric_representation_item <= representation_item <- representation.items[i] representation {representation => presentation_representation => presentation_view} representation.items[i] -> representation_item => mapped_item => camera_image => (camera_image_2d_with_scale) (camera_image_3d_with_scale)
SURFACE_APPEARANCE	(surface_style_usage) (surface_style_usage)	46 517		
side	(surface_style_usage.side) (surface_style_usage.side)	46 517		

Table 21 – Mapping table shaded_presentation UoF (P3) (continued)

Application element	AIM element	Source	Rules	Reference path
surface_ appearance to surface_ appearance_ element (as element)	PATH			surface_style_usage surface_style_usage.style -> surface_side_style_select surface_side_style_select = surface_side_style surface_side_style.styles[i] -> surface_style_element_select (surface_style_element_select = surface_style_boundary surface_style_boundary) (surface_style_element_select = surface_style_parameter_line surface_style_parameter_line) (surface_style_element_select = surface_style_control_grid surface_style_control_grid) (surface_style_element_select = surface_style_fill_area surface_style_fill_area) (surface_style_element_select = surface_style_segmentation_curve surface_style_segmentation_curve) (surface_style_element_select = surface_style_silhouette surface_style_silhouette) (surface_style_element_select = surface_style_rendering surface_style_rendering) (surface_style_element_select = surface_style_rendering surface_style_rendering => surface_style_rendering_with_properties)
SURFACE_ APPEARANCE_ ELEMENT	(surface_style_ boundary) (surface_style_ parameter_line) (surface_style_ control_grid) (surface_style_fill_ area) (surface_style_ segmentation_curve) (surface_style_ silhouette)	517 517 517 517 517 517		{surface_style_rendering_with_properties surface_style_rendering_with_properties.properties[i] -> rendering_properties_select (rendering_properties_select = surface_style_reflectance_ambient surface_style_reflectance_ambient) (rendering_properties_select = surface_style_reflectance_ambient surface_style_reflectance_ambient => surface_style_reflectance_ambient_diffuse) (rendering_properties_select = surface_style_reflectance_ambient

Table 21 – Mapping table shaded_presentation UoF (P3) (continued)

Application element	AIM element	Source	Rules	Reference path
	(surface_style_rendering) (surface_style_rendering_with_properties)	46 46		surface_style_reflectance_ambient => surface_style_reflectance_ambient_diffuse => surface_style_reflectance_ambient_diffuse_specular) (rendering_properties_select = surface_style_transparent surface_style_transparent)}
surface_appearance_element to visual_appearance (as surface_condition)	PATH			(surface_style_boundary surface_style_element_select = surface_style_boundary) (surface_style_parameter_line surface_style_element_select = surface_style_parameter_line) (surface_style_control_grid surface_style_element_select = surface_style_control_grid) (surface_style_fill_area surface_style_element_select = surface_style_fill_area) (surface_style_segmentation_curve surface_style_element_select = surface_style_segmentation_curve) (surface_style_silhouette surface_style_element_select = surface_style_silhouette) (surface_style_rendering surface_style_element_select = surface_style_rendering) (surface_style_rendering_with_properties <= surface_style_rendering surface_style_element_select = surface_style_rendering) surface_style_element_select <- surface_side_style.styles[i] surface_side_style surface_side_style_select = surface_side_style surface_side_style_select <- surface_style_usage.style surface_style_usage presentation_style_select = surface_style_usage presentation_style_select <- presentation_style_assignment.styles[i] presentation_style_assignment <- styled_item.styles[i]

Table 21 – Mapping table shaded_presentation UoF (P3) (concluded)

Application element	AIM element	Source	Rules	Reference path
				styled_item styled_item.item ->
representation_item <- representation.items[i] representation {[representation representation.description = 'visual appearance'] [representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.definition ->
represented_definition represented_definition = property_definition property_definition => surface_condition]}

Table 22 – Mapping table item_property UoF (PR1)

Application element	AIM element	Source	Rules	Reference path
CENTRE_OF_MASS	property_definition	41		{property_definition.name = 'centre of mass'}
centre_point	point	42		property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => {representation_item.name = 'centre of mass'} geometric_representation_item => point
description	property_definition. description	41		
value_ interpretation	representation_ context.context_type	43		property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.context_of_items -> representation_context representation_context.context_type {(representation_context.context_type) (representation_context.context_type = 'required') (representation_context.context_type= 'calculated') (representation_context.context_type= 'measured') (representation_context.context_type= 'estimated')}
centre_of_mass to multi_ language_string (as description)	PATH		149	property_definition multi_language_attribute_item = property_definition multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <=

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
				multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
centre_of_mass to cartesian_ coordinate_space (as is_defined_ in)	PATH			property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.context_of_items -> representation_context => geometric_representation_context
DATA_ENVIRONMENT	data_environment	45		
description	data_environment. description	45		
name	data_environment. name	45		
data_environment to multi_ language_string (as description)	PATH		99	data_environment multi_language_attribute_item = data_environment multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
DEFAULT_SETTING_ ASSOCIATION	(property_definition_ representation) (representation_ relationship)	41 43		({property_definition_representation property_definition_representation.name = 'default setting'}) ({representation_relationship representation_relationship.name = 'default setting'})
default_setting_ association to digital_document (as default_ context)	PATH		27	representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_2 -> representation => externally_defined_representation digital_document_reference_item = externally_defined_representation digital_document_reference_item <-

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
				digital_document_reference.items[1] digital_document_reference <= document_reference {document_reference.role -> document_role document_role.name = 'external definition'} document_reference.assigned_document -> document {document.kind -> document_type document_type.product_data_type = 'digital'} document
default_setting_association to drawing_sheet (as default_context)	PATH			representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_2 -> representation => presentation_representation => presentation_area => drawing_sheet_revision
default_setting_association to item_shape (as default_context)	PATH			property_definition_representation {property_definition_representation.name = 'default setting'} property_definition_representation.definition -> represented_definition = property_definition property_definition => product_definition_shape
default_setting_association to view_area (as default_context)	PATH			representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_2 -> representation => presentation_representation => (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
default_setting_ association to default_ thickness (as default_ value)	PATH			(property_definition_representation {property_definition_representation.name = 'default setting'} property_definition_representation.used_representation ->) (representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_1 ->) representation {representation.description = 'default thickness'}
DEFAULT_ THICKNESS	representation	43		{representation [representation.name = 'default thickness'] [representation.context_of_items -> representation_context representation_context.context_type = 'default setting']}
description	representation. description	43		
default_ thickness to multi_language_ string (as description)	PATH		153	representation {representation.name = 'default thickness'} multi_language_attribute_item = representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
default_ thickness to value_with_unit (as thickness_ value)	PATH			representation {representation.name = 'default thickness'} representation.items[i] -> representation_item => measure_representation_item <= measure_with_unit
GENERAL_ITEM_ PROPERTY	property_definition	41		
description	property_definition. description	41		
property_type	property_definition. name	41		

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
general_item_ property to multi_language_ string (as description)	PATH		149	property_definition multi_language_attribute_item = property_definition multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
general_item_ property to external_model (as property_ value)	PATH			property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => externally_defined_representation
general_item_ property to geometric_model (as property_ value)	PATH			property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => shape_representation
general_item_ property to property_value (as property_ value)	PATH			property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => property_value_representation

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
general_item_ property to styled_model (as property_ value)	PATH			property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->* representation =>* (draughting_model) (mechanical_design_geometric_presentation_representation)
ITEM_COST_ PROPERTY	property_definition	41		{property_definition.name = 'cost property'}
item_cost_ property to property_value (as cost_ property_value)	PATH			property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->* representation =>* property_value_representation
ITEM_PROPERTY	property_definition	41		
item_property to design_ constraint (as described_ object)	PATH			property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->* representation <- representation_relationship.rep_1 representation_relationship {representation_relationship.name = 'design constraint'} representation_relationship.rep_2 ->* representation {representation.description = 'design constraint'}

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
item_property to design_ discipline_item_ definition (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition
item_property to feature_ definition (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = characterized_object characterized_object => feature_definition
item_property to general_feature (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect
item_property to item_definition_ instance_ relationship (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship
item_property to item_definition_ relationship (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship
item_property to item_instance (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = characterized_product_definition (characterized_product_definition = product_definition product_definition => product_definition_occurrence) (characterized_product_definition = product_definition_relationship product_definition_relationship => product_definition_usage =>

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
				assembly_component_usage)
item_property to item_instance_ relationship (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}
item_property to item_shape (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = shape_definition shape_definition = product_definition_shape product_definition_shape
item_property to item_solution (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition => solution_based_product_definition
item_property to physical_ instance (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition
item_property to product_class (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = characterized_object characterized_object => product_class

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
item_property to product_ component (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = characterized_object characterized_object => product_component
item_property to product_function (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition {product_definition [product_definition.frame_of_reference ->product_definition_context <=application_context_element application_context_element.name = 'functional definition'] [product_definition.formation ->product_definition_formation product_definition_formation.of_product ->product product.frame_of_reference[i] ->product_context <=application_context_element application_context_element.name = 'functionality']}
item_property to shape_aspect (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect
item_property to shape_aspect_ relationship (as described_ object)	PATH			property_definition property_definition.definition ->characterized_definition = shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
item_property to thread_feature (as described_object)	PATH			property_definition property_definition.definition ->characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect => thread_occurrence
item_property to transition_feature (as described_object)	PATH			property_definition property_definition.definition ->characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect => transition_feature
ITEM_QUALITY_PROPERTY	property_definition	41		{property_definition.name = 'quality measure property'}
description	property_definition.description	41		
item_quality_property to multi_language_string (as description)	PATH		149	property_definition multi_language_attribute_item = property_definition multi_language_attribute_item <-applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <=multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
item_quality_property to property_value (as quality_property_value)	PATH			property_definition <-property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->representation => property_value_representation
MASS	property_definition	41		{property_definition.name = 'mass'}
mass to property_value (as mass_property_value)	PATH			property_definition <-property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->representation =>

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_value_representation
MATERIAL	material_designation	45		
name	material_designation. name	45		
material to design_ constraint (as described_ object)	PATH			material_designation <- material_designation_characterization.designation material_designation_characterization material_designation_characterization.property -> characterized_material_property = material_property_representation material_property_representation <= property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation <- representation_relationship.rep_1 representation_relationship {representation_relationship.name = 'design constraint'} representation_relationship.rep_2 -> representation {representation.description = 'design constraint'}
material to design_ discipline_item_ definition (as described_ object)	PATH			material_designation material_designation.definitions[i] -> characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition
material to feature_ definition (as described_ object)	PATH			material_designation material_designation.definitions[i] -> characterized_definition = characterized_object characterized_object => feature_definition

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
material to general_feature (as described_object)	PATH			material_designation material_designation.definitions[i] ->characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect
material to item_definition_instance_relationship (as described_object)	PATH			material_designation material_designation.definitions[i] ->characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship
material to item_definition_relationship (as described_object)	PATH			material_designation material_designation.definitions[i] ->characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship
material to item_instance (as described_object)	PATH			material_designation material_designation.definitions[i] ->characterized_definition = characterized_product_definition (characterized_product_definition = product_definition product_definition => product_definition_occurrence) (characterized_product_definition = product_definition_relationship product_definition_relationship => product_definition_usage => assembly_component_usage)
material to item_instance_relationship (as described_object)	PATH			material_designation material_designation.definitions[i] ->characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
material to item_ shape (as described_ object)	PATH			material_designation material_designation.definitions[i] -> characterized_definition = shape_definition shape_definition = product_definition_shape product_definition_shape
material to item_ solution (as described_ object)	PATH			material_designation material_designation.definitions[i] -> characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition => solution_based_product_definition
material to physical_ instance (as described_ object)	PATH			material_designation material_designation.definitions[i] -> characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition
material to product_class (as described_ object)	PATH			material_designation material_designation.definitions[i] -> characterized_definition = characterized_object characterized_object => product_class
material to product_ component (as described_ object)	PATH			material_designation material_designation.definitions[i] -> characterized_definition = characterized_object characterized_object => product_component
material to product_function (as described_ object)	PATH			material_designation material_designation.definitions[i] -> characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition {product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition']

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
				[product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}
material to shape_aspect (as described_object)	PATH			material_designation material_designation.definitions[i] -> characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect
material to shape_aspect_relationship (as described_object)	PATH			material_designation material_designation.definitions[i] -> characterized_definition = shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship
material to thread_feature (as described_object)	PATH			material_designation material_designation.definitions[i] -> characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect => thread_occurrence
material to transition_feature (as described_object)	PATH			material_designation material_designation.definitions[i] -> characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect => transition_feature
material to material_property (as material_property)	PATH			material_designation <- material_designation_characterization.designation material_designation_characterization material_designation_characterization.property -> characterized_material_property = material_property_representation material_property_representation <=

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation property_definition_representation.definition ->\ property_definition =>\ material_property
MATERIAL_ PROPERTY	material_property	45		
description	property_definition. description	41		material_property <= property_definition property_definition.description
property_name	property_definition. name	41		material_property <= property_definition property_definition.name
material_ property to multi_language_ string (as description)	PATH		119	material_property multi_language_attribute_item = material_property multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'property_definition.description'}
material_ property to material_ property_ representation (as material_ property_value)	PATH			material_property <= property_definition <- property_definition_representation.definition property_definition_representation =>\ material_property_representation
MATERIAL_ PROPERTY_ REPRESENTATION	material_property_ representation	45		

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
material_ property_ representation to data_ environment (as environment_ condition)	PATH			material_property_representation material_property_representation.dependent_environment ->data_environment
material_ property_ representation to property_ value (as property_ value)	PATH			material_property_representation <=representation representation.items[i] ->representation_item =>mapped_item mapped_item.mapping_source ->representation_map representation_map.mapped_representation ->representation =>property_value_representation
MATERIAL_SIDE_ SPECIFICATION	representation	43		{representation.name = 'material side specification'}
direction	direction	42		representation representation.items[1] ->representation_item => geometric_representation_item =>direction
material_side_ specification to cartesian_ coordinate_space (as is_defined_ in)	PATH			representation representation.context_of_items ->representation_context => geometric_representation_context
material_side_ specification to point_on_surface (as position)	PATH			representation representation.items[2] ->representation_item => geometric_representation_item =>point => point_on_surface

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
MOMENTS_OF_ INERTIA	property_definition	41		{property_definition.name = 'moments of inertia'}
value_ interpretation	representation_ context.context_type	43		property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.context_of_items -> representation_context representation_context.context_type {(representation_context.context_type) (representation_context.context_type = 'required') (representation_context.context_type= 'calculated') (representation_context.context_type= 'measured') (representation_context.context_type= 'estimated')}
moments_of_ inertia to value_ with_unit (as inertia_ value)	PATH			property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {representation.name = 'moments of inertia matrix'} [representation.items[i] -> representation_item => {representation_item.name = 'm1 1'}[1)] measure_representation_item <=] [representation.items[i] -> representation_item => {representation_item.name = 'm1 2'}[2)] measure_representation_item <=] [representation.items[i] -> representation_item => {representation_item.name = 'm1 3'}[3)]

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
				measure_representation_item <=] [representation.items[i] -> representation_item => {representation_item.name = 'm2 1'}[4)] measure_representation_item <=] [representation.items[i] -> representation_item => {representation_item.name = 'm2 2'}[5)] measure_representation_item <=] [representation.items[i] -> representation_item => {representation_item.name = 'm2 3'}[6)] measure_representation_item <=] [representation.items[i] -> representation_item => {representation_item.name = 'm3 1'}[7)] measure_representation_item <=] [representation.items[i] -> representation_item => {representation_item.name = 'm3 2'}[8)] measure_representation_item <=] [representation.items[i] -> representation_item => {representation_item.name = 'm3 3'}[9)] measure_representation_item <=] measure_with_unit
moments_of_ inertia to cartesian_ coordinate_space (as is_defined_ in)	PATH			property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.context_of_items -> representation_context => geometric_representation_context

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
PROPERTY	(property_ definition) (action_property)	41 49		
PROPERTY_CHANGE	property_change	214		property_change <= property_definition
description	property_definition. description	41		(property_change <= property_definition property_definition.description)
id	property_definition. name	41		(property_change <= property_definition property_definition.name)
property_change to property (as added_ property)	PATH			property_change <= property_definition <- property_definition_relationship.relating_property_definition property_definition_relationship {property_definition_relationship.name = 'added property'} property_definition_relationship.related_property_definition -> property_definition
property_change to property (as deleted_ property)	PATH			property_change <= property_definition <- property_definition_relationship.relating_property_definition property_definition_relationship {property_definition_relationship.name = 'deleted property'} property_definition_relationship.related_property_definition -> property_definition
property_change to multi_ language_string (as description)	PATH		148	property_change multi_language_attribute_item = property_change multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'property_definition.description'}

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
property_change to change (as is_describing)	PATH		148	property_change <= property_definition property_definition.definition -> characterized_definition (characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship) (characterized_definition = shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship)
PROPERTY_RELATIONSHIP	property_definition_relationship	45		
description	property_definition_relationship.description	45		
relation_type	property_definition_relationship.name	45		{(property_definition_relationship.name) (property_definition_relationship.name = 'dependency') (property_definition_relationship.name = 'hierachy')}
property_relationship to multi_language_string (as description)	PATH			property_definition_relationship multi_language_attribute_item = property_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
property_relationship to property (as related)	PATH			property_definition_relationship property_definition_relationship.related_property_definition -> property_definition
property_relationship to property (as relating)	PATH			property_definition_relationship property_definition_relationship.relating_property_definition -> property_definition

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
PROPERTY_VALUE	property_value_representation	214		property_value_representation <= representation
lower_limit	(measure_representation_item) (descriptive_representation_item)	45 45		property_value_representation <= representation representation.items[i] -> representation_item => {representation_item.name = 'lower'} (measure_representation_item {measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component -> (unit = named_unit) (unit = derived_unit)]}) (descriptive_representation_item)
name	representation.name	43		property_value_representation <= representation representation.name
specified_value	descriptive_representation_item	45		property_value_representation <= representation representation.items[i] -> representation_item => {representation_item.name = 'specified value'} descriptive_representation_item
upper_limit	(measure_representation_item) (descriptive_representation_item)	45 45		property_value_representation <= representation representation.items[i] -> representation_item => {representation_item.name = 'upper'} (measure_representation_item {measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component -> (unit = named_unit) (unit = derived_unit)]})

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
				(descriptive_representation_item)
value_ interpretation	representation_ context.context_type	43		property_value_representation <= representation representation.context_of_items -> representation_context representation_context.context_type {(representation_context.context_type) (representation_context.context_type = 'required') (representation_context.context_type= 'calculated') (representation_context.context_type= 'measured') (representation_context.context_type= 'estimated')}
property_value to multi_ language_string (as specified_ value)	PATH		101	property_value_representation <= representation representation.items[i] -> representation_item => {representation_item.name = 'specified value'} descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
property_value to value_with_ unit (as specified_ value)	PATH			property_value_representation <= representation representation.items[i] -> representation_item => {representation_item.name = 'specified value'} measure_representation_item <= measure_with_unit
RECYCLABILITY	property_definition	41		{property_definition.name = 'recyclability'}
description	property_definition. description	41		

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
recyclability to multi_language_ string (as description)	PATH		149	property_definition multi_language_attribute_item = property_definition multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
recyclability to property_value (as recycling_ class_value)	PATH			property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => property_value_representation
SHAPE_DEPENDENT_ PROPERTY	property_definition	41		
shape_dependent_ property to item_ shape (as described_ object)	PATH			property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition = product_definition_shape product_definition_shape
shape_dependent_ property to shape_aspect (as described_ object)	PATH			property_definition property_definition.definition -> characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect
THICKNESS	property_definition	41		{property_definition.name = 'thickness'}
description	property_definition. description	41		
thickness to multi_language_ string (as description)	PATH		149	property_definition multi_language_attribute_item = property_definition multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
				{applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
thickness to material_side_ specification (as material_ side)	PATH			property_definition {property_definition.name = 'thickness'} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {representation.name = 'material side specification'}
thickness to property_value (as thickness_ value)	PATH			property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation => property_value_representation
VALUE_WITH_UNIT	measure_with_unit	41	35	{measure_with_unit => (length_measure_with_unit) (mass_measure_with_unit) (time_measure_with_unit) (electric_current_measure_with_unit) (thermodynamic_temperature_measure_with_unit) (amount_of_substance_measure_with_unit) (luminous_intensity_measure_with_unit) (plane_angle_measure_with_unit) (solid_angle_measure_with_unit) (area_measure_with_unit) (volume_measure_with_unit) (ratio_measure_with_unit)}
unit_component	(named_unit) (derived_unit)	41 41	25	measure_with_unit measure_with_unit.unit_component -> (unit = named_unit

Table 22 – Mapping table item_property UoF (PR1) (continued)

Application element	AIM element	Source	Rules	Reference path
				named_unit {named_unit => [(si_unit) (conversion_based_unit) (expression_conversion_based_unit) (context_dependent_unit)] [(length_unit) (mass_unit) (time_unit) (electric_current_unit) (thermodynamic_temperature_unit) (amount_of_substance_unit) (luminous_intensity_unit) (plane_angle_unit) (solid_angle_unit) (area_unit) (volume_unit) (ratio_unit)]}) (unit = derived_unit derived_unit)
value_component	measure_value	41		measure_with_unit measure_with_unit.value_component -> measure_value {(measure_value = length_measure length_measure) (measure_value = mass_measure mass_measure) (measure_value = time_measure time_measure) (measure_value = electric_current_measure electric_current_measure) (measure_value = thermodynamic_temperature_measure thermodynamic_temperature_measure) (measure_value = amount_of_substance_measure amount_of_substance_measure) (measure_value = luminous_intensity_measure

Table 22 – Mapping table item_property UoF (PR1) (concluded)

Application element	AIM element	Source	Rules	Reference path
				luminous_intensity_measure) (measure_value = plane_angle_measure plane_angle_measure) (measure_value = solid_angle_measure solid_angle_measure) (measure_value = area_measure area_measure) (measure_value = volume_measure volume_measure) (measure_value = ratio_measure ratio_measure) (measure_value = parameter_value parameter_value) (measure_value = numeric_measure numeric_measure) (measure_value = context_dependent_measure context_dependent_measure) (measure_value = descriptive_measure descriptive_measure) (measure_value = positive_length_measure positive_length_measure) (measure_value = positive_plane_angle_measure positive_plane_angle_measure) (measure_value = positive_ratio_measure positive_ratio_measure) (measure_value = count_measure count_measure)}

NOTE – The following note numbers correspond to the footnotemarks listed in the table above:

1, 2, 3, 4, 5, 6, 7, 8, 9 The first number specifies the row, the second number the column of the element in the moments_of_inertia_matrix.

Table 23 – Mapping table product_management_data UoF (S1)

Application element	AIM element	Source	Rules	Reference path
ALIAS_ IDENTIFICATION	applied_ identification_ assignment	214		applied_identification_assignment <= identification_assignment
alias_id	identification_ assignment.assigned_ id	41	33	applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'alias id'} identification_assignment.assigned_id
alias_version_id	identification_ assignment.assigned_ id	41	33	applied_identification_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'alias version id'} identification_assignment.assigned_id
description	identification_role. description	41	33	applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.description
alias_ identification to organization (as alias_scope)	PATH			applied_identification_assignment organization_item = applied_identification_assignment organization_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment.role -> organization_role organization_role.name = 'alias scope'} organization_assignment.assigned_organization -> organization

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
alias_ identification to multi_ language_string (as description)	PATH			applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role multi_language_attribute_item = identification_role multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
alias_ identification to approval_ status (as is_applied_ to)	PATH			applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = approval_status approval_status
alias_ identification to classification_ system (as is_applied_ to)	PATH			applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item (identification_item = classification_system classification_system) (identification_item = externally_defined_classification_system externally_defined_classification_system)

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
alias_ identification to document_type (as is_applied_ to)	PATH			applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = document_type document_type
alias_ identification to external_ document (as is_applied_ to)	PATH		173	(applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item (identification_item = document document) (identification_item = physically_modelled_product_definition physically_modelled_product_definition)) ([applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item (identification_item = document document) (identification_item = physically_modelled_product_definition physically_modelled_product_definition)]

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				[applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias version id'} applied_identification_assignment.items[1] -> identification_item identification_item = version_assignment version_assignment.items[1] -> version_item (version_item = document document) (version_item = physically_modelled_product_definition physically_modelled_product_definition)])
alias_ identification to geometric_ model (as is_applied_ to)	PATH		181	(applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = shape_representation) ([applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = shape_representation] [applied_identification_assignment

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				{applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias version id'} applied_identification_assignment.items[1] -> identification_item identification_item = version_assignment version_assignment version_assignment.items[1] -> version_item version_item = shape_representation]) shape_representation
alias_ identification to item_instance (as is_applied_ to)	PATH			applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item (identification_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (identification_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <=

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})
alias_ identification to item_solution (as is_applied_ to)	PATH		182	(applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = solution_based_product_definition) ([applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = solution_based_product_definition] [applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias version id'} applied_identification_assignment.items[1] -> identification_item identification_item = version_assignment version_assignment version_assignment.items[1] -> version_item

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				version_item = solution_based_product_definition]) solution_based_product_definition
alias_ identification to physical_ instance (as is_applied_ to)	PATH			applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}
alias_ identification to product_class (as is_applied_ to)	PATH		176	(applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = product_class) ([applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = product_class]

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				[applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias version id'} applied_identification_assignment.items[1] -> identification_item identification_item = version_assignment version_assignment version_assignment.items[1] -> version_item = product_class]) product_class
alias_ identification to product_ component (as is_applied_ to)	PATH		177	(applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = product_component) ([applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = product_component] [applied_identification_assignment {applied_identification_assignment <=

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				identification_assignment identification_assignment.role ->
identification_role identification_role.name = 'alias version id'} applied_identification_assignment.items[1] ->
identification_item identification_item = version_assignment version_assignment version_assignment.items[1] ->
version_item version_item = product_component]) product_component
alias_ identification to product_ function (as is_applied_ to)	PATH		179	(applied_identification_assignment {applied_identification_assignment <=
identification_assignment identification_assignment.role ->
identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] ->
identification_item identification_item = product_definition) ([applied_identification_assignment {applied_identification_assignment <=
identification_assignment identification_assignment.role ->
identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] ->
identification_item identification_item = product_definition] [applied_identification_assignment {applied_identification_assignment <=
identification_assignment identification_assignment.role ->
identification_role

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				identification_role identification_role.name = 'alias version id'} applied_identification_assignment.items[1] -> identification_item identification_item = version_assignment version_assignment version_assignment.items[1] -> version_item version_item = product_definition]) product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}
alias_ identification to security_ level (as is_applied_ to)	PATH			applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = security_classification_level security_classification_level

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
alias_ identification to specification (as is_applied_ to)	PATH		178	(applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)) ([applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)] [applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias version id'} applied_identification_assignment.items[1] -> identification_item identification_item = version_assignment version_assignment

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				version_assignment.items[1] -> version_item version_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)])
alias_ identification to specification_ category (as is_applied_ to)	PATH			applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item identification_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category)
alias_ identification to styled_model (as is_applied_ to)	PATH		171,172	(applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item (identification_item = draughting_model draughting_model) (identification_item = \ mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)) ([applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role ->

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				identification_role identification_role.name = 'alias id'} applied_identification_assignment.items[1] -> identification_item (identification_item = draughting_model draughting_model) (identification_item = mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)] [applied_identification_assignment {applied_identification_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'alias version id'} applied_identification_assignment.items[1] -> identification_item identification_item = version_assignment version_assignment version_assignment.items[1] -> version_item (version_item = draughting_model draughting_model) (version_item = mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)])
APPLICATION_ CONTEXT	(product_definition_ context) (configuration_item_ context)	41 41		
application_ domain	application_context. name	41		(product_definition_context <=) (configuration_item_context <=) application_context_element application_context_element.frame_of_reference[1] -> application_context application_context.name

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
description	application_context. description			(product_definition_context <=) (configuration_item_context <=) application_context_element application_context_element.frame_of_reference[1] -> application_context application_context.description
life_cycle_stage	(product_definition_ context.life_cycle_ stage) (configuration_item_ context.usage)	41 41		
application_ context to multi_ language_string (as description)	PATH		86	(product_definition_context <=) (configuration_item_context <=) application_context_element application_context_element.frame_of_reference[1] -> application_context multi_language_attribute_item = application_context multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
APPROVAL	approval	41	15	{approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment}
description	approval.level	41		
approval to date_ time (as actual_date)	PATH		74,14	approval <- approval_date_time.dated_approval approval_date_time {approval_date_time.role -> approval_date_time_role approval_date_time_role.name = 'actual'} approval_date_time.date_time -> date_time_select

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				(date_time_select = date date => calendar_date) (date_time_select = date_and_time date_and_time)
approval to approval_status (as approval_ status)	PATH			approval approval.status ->approval_status
approval to multi_language_ string (as description)	PATH		87	approval multi_language_attribute_item = approval multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'level'}
approval to activity (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->approval_item approval_item = action action
approval to activity_element (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->approval_item approval_item = applied_action_assignment applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role ->

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_role action_role.description = 'activity element'}
approval to alternate_item_ relationship (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->\ approval_item approval_item = alternate_product_relationship alternate_product_relationship
approval to assembly_ substitute_ relationship (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->\ approval_item approval_item = assembly_component_usage_substitute assembly_component_usage_substitute
approval to class_category_ association (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->\ approval_item approval_item = product_concept_feature_category_usage product_concept_feature_category_usage {product_concept_feature_category_usage <= group_assignment group_assignment.role ->\ group_role (group_role.name = 'mandatory category usage') (group_role.name = 'optional category usage')}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
approval to class_condition_ association (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = product_concept_feature_association product_concept_feature_association
approval to class_function_ association (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = configuration_design configuration_design {configuration_design.configuration -> configuration_item configuration_item.description = 'class function definition'}
approval to class_inclusion_ association (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = product_concept_feature_association product_concept_feature_association {product_concept_feature_association.name = 'inclusion association'}
approval to class_ specification_ association (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = product_concept_feature_association product_concept_feature_association

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
approval to configuration (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->> approval_item [approval_item = configured_effective_item_assignment configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role ->> effectivity_role effectivity_role.name = 'configuration'}] [approval_item = configured_effective_context_assignment configured_effective_context_assignment {configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role ->> effectivity_role effectivity_role.name = 'configuration'}]
approval to design_ constraint (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->> approval_item approval_item = representation representation {representation.description = 'design constraint'}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
approval to design_ discipline_item_ definition (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = product_definition product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
approval to drawing (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = auto_design_drawing_revision auto_design_drawing_revision
approval to external_ document (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item (approval_item = document document) (approval_item = physically_modelled_product_definition physically_modelled_product_definition)
approval to geometric_model (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				approval_item = shape_representation shape_representation
approval to item_ definition_ instance_ relationship (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->\ approval_item approval_item = product_definition_relationship product_definition_relationship {[(product_definition_relationship.related_product_definition ->\ product_definition => product_definition_occurrence) (product_definition_relationship => {(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')} product_definition_usage => assembly_component_usage)] [product_definition_relationship.relating_product_definition ->\ product_definition product_definition.frame_of_reference ->\ product_definition_context <= application_context_element application_context_element.name = 'definitional template']}
approval to item_ definition_ relationship (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->\ approval_item approval_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->]

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				[product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
approval to item_ instance (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item (approval_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (approval_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
approval to item_ instance_ relationship (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item (approval_item = product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}) (approval_item = product_definition_substitute product_definition_substitute)
approval to item_ solution (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = solution_based_product_definition solution_based_product_definition
approval to item_ version (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = product_definition_formation product_definition_formation
approval to material (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				approval_item = material_designation material_designation
approval to physical_ instance (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->\ approval_item approval_item = product_definition product_definition {[product_definition.frame_of_reference ->\ product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}
approval to physical_ instance_usage (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->\ approval_item approval_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'physical occurrence usage'}
approval to process_plan (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->\ approval_item approval_item = process_plan process_plan

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
approval to product_class (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = product_class product_class
approval to product_ component (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = product_component product_component
approval to product_function (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
approval to project (as is_applied_to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = project project
approval to simulation_information (as is_applied_to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = kinematic_property_definition kinematic_property_definition
approval to specification (as is_applied_to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)
approval to specification_category (as is_applied_to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category)

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
approval to specification_ expression (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = conditional_concept_feature conditional_concept_feature {conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'expression'}
approval to specification_ inclusion (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item approval_item = conditional_concept_feature conditional_concept_feature {[conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'inclusion'] [conditional_concept_feature.condition -> concept_feature_relationship_with_condition concept_feature_relationship_with_condition.conditional_operator = 'implication']}
approval to styled_model (as is_applied_ to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] -> approval_item (approval_item = draughting_model draughting_model) (approval_item = mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
approval to view_area (as is_applied_to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->approval_item approval_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)
approval to work_order (as is_applied_to)	PATH			approval <- approval_assignment.assigned_approval approval_assignment => applied_approval_assignment applied_approval_assignment.items[i] ->approval_item approval_item = action_directive action_directive
approval to date_and_person_or_organization (as is_approved_by)	PATH		5,16	approval <- approval_person_organization.authorized_approval approval_person_organization {[approval_person_organization.role ->approver_role approver_role.name = 'approver'] [(date_and_time_item = approval_person_organization date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.role ->date_time_role date_time_role.name = 'actual') (date_item = approval_person_organization date_item <- applied_date_assignment.items[i] applied_date_assignment <=

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				date_assignment date_assignment.role ->& date_role date_role.name = 'actual')]}
approval to date_time (as nominal_date)	PATH		74,14	approval <- approval_date_time.dated_approval approval_date_time {approval_date_time.role ->& approval_date_time_role approval_date_time_role.name = 'nominal'} approval_date_time.date_time ->& date_time_select (date_time_select = date date =>& calendar_date) (date_time_select = date_and_time date_and_time)
APPROVAL_RELATIONSHIP	approval_relationship	41		
relation_type	approval_relationship.name	41		
approval_relationship to approval (as related)	PATH			approval_relationship approval_relationship.related_approval ->& approval
approval_relationship to approval (as relating)	PATH			approval_relationship approval_relationship.relating_approval ->& approval
APPROVAL_STATUS	approval_status	41		
name	approval_status.name	41		

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
approval_status to classification_system (as used_classification_system)	PATH		76	approval_status group_member_item = approval_status group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment {group_assignment.role -> group_role group_role.name = 'classified member'} group_assignment.assigned_group -> group => (classification_system) (externally_defined_classification_system)
DATE_AND_PERSON_ASSIGNMENT	(applied_date_and_person_assignment) (applied_date_and_person_and_organization_assignment) (applied_date_and_organization_assignment)	214 214 214		(applied_date_and_person_assignment <= person_assignment) (applied_date_and_person_and_organization_assignment <= person_and_organization_assignment) (applied_date_and_organization_assignment <= organization_assignment)
role	(person_role.name) (person_and_organization_role.name) (organization_role.name)	41 41 41	39,37	(applied_date_and_person_assignment <= person_assignment person_assignment.role -> person_role person_role.name {(person_role.name) (person_role.name = 'creation') (person_role.name = 'update')}) (applied_date_and_person_and_organization_assignment <= person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				{(person_and_organization_role.name) (person_and_organization_role.name = 'creation') (person_and_organization_role.name = 'update')}) (applied_date_and_organization_assignment <= organization_assignment organization_assignment.role -> organization_role organization_role.name {(organization_role.name) (organization_role.name = 'creation') (organization_role.name = 'update')})
date_and_person_ assignment to date_and_person_ or_organization (as assigned_ date_and_person)	PATH			([applied_date_and_person_assignment (date_item = applied_date_and_person_assignment date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment {date_assignment.role -> date_role date_role.name = 'actual'} date_assignment.assigned_date -> date => calendar_date) (date_and_time_item = applied_date_and_person_assignment date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment.role -> date_time_role date_time_role.name = 'actual'} date_and_time_assignment.assigned_date_and_time -> date_and_time)] [applied_date_and_person_assignment <= person_assignment

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				person_assignment.assigned_person -> person]) ([applied_date_and_person_and_organization_assignment (date_item = applied_date_and_person_and_organization_assignment date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment {date_assignment.role -> date_role date_role.name = 'actual'} date_assignment.assigned_date -> date => calendar_date) (date_and_time_item = applied_date_and_person_and_organization_assignment date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment.role -> date_time_role date_time_role.name = 'actual'} date_and_time_assignment.assigned_date_and_time -> date_and_time)] [applied_date_and_person_and_organization_assignment <= person_and_organization_assignment person_and_organization_assignment.assigned_person_and_organization -> person_and_organization]) ([applied_date_and_organization_assignment (date_item = applied_date_and_organization_assignment date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				{date_assignment.role -> date_role date_role.name = 'actual'} date_assignment.assigned_date -> date => calendar_date) (date_and_time_item = applied_date_and_organization_assignment date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment.role -> date_time_role date_time_role.name = 'actual'} date_and_time_assignment.assigned_date_and_time -> date_and_time)] [applied_date_and_organization_assignment <= organization_assignment organization_assignment.assigned_organization -> organization])
date_and_person_ assignment to activity (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = action) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = action) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = action) action

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_and_person_ assignment to activity_element (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = applied_action_assignment) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = applied_action_assignment) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = applied_action_assignment) applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'}
date_and_person_ assignment to assembly_ substitute_ relationship (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = assembly_component_usage_substitute) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = \ assembly_component_usage_substitute) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = assembly_component_usage_substitute) assembly_component_usage_substitute

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_and_person_ assignment to class_category_ association (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_concept_feature_category_usage) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = \ product_concept_feature_category_usage) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = product_concept_feature_category_usage) product_concept_feature_category_usage {product_concept_feature_category_usage <= group_assignment group_assignment.role -> group_role (group_role.name = 'mandatory category usage') (group_role.name = 'optional category usage')}
date_and_person_ assignment to class_condition_ association (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_concept_feature_association) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = product_concept_feature_association) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = product_concept_feature_association) product_concept_feature_association

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_and_person_ assignment to class_function_ association (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = configuration_design) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = configuration_design) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = configuration_design) configuration_design {configuration_design.configuration -> configuration_item configuration_item.description = 'class function definition'}
date_and_person_ assignment to class_inclusion_ association (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_concept_feature_association) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = product_concept_feature_association) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = product_concept_feature_association) product_concept_feature_association {product_concept_feature_association.name = 'inclusion association'}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_and_person_ assignment to class_ specification_ association (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_concept_feature_association) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = product_concept_feature_association) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = product_concept_feature_association) product_concept_feature_association
date_and_person_ assignment to configuration (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item [date_and_person_item = configured_effective_item_assignment configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}] [date_and_person_item = configured_effective_context_assignment configured_effective_context_assignment {configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}]) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item [date_and_person_and_organization_item = \ configured_effective_item_assignment

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}] [date_and_person_and_organization_item = \ configured_effective_context_assignment configured_effective_context_assignment {configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}]) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item [date_and_organization_item = configured_effective_item_assignment configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}] [date_and_organization_item = configured_effective_context_assignment configured_effective_context_assignment {configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}])

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_and_person_ assignment to design_ constraint (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = representation) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = representation) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = representation) representation {representation.description = 'design constraint'}
date_and_person_ assignment to design_ discipline_item_ definition (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_definition) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = product_definition) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = product_definition) product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_and_person_ assignment to drawing (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = auto_design_drawing_revision) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = auto_design_drawing_revision) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = auto_design_drawing_revision) auto_design_drawing_revision
date_and_person_ assignment to external_ document (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item (date_and_person_item = document document) (date_and_person_item = physically_modelled_product_definition physically_modelled_product_definition)) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item (date_and_person_and_organization_item = document document) (date_and_person_and_organization_item = \ physically_modelled_product_definition physically_modelled_product_definition)) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item (date_and_organization_item = document document) (date_and_organization_item = physically_modelled_product_definition

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				physically_modelled_product_definition))
date_and_person_ assignment to geometric_model (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = shape_representation) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = shape_representation) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = shape_representation) shape_representation
date_and_person_ assignment to item (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = product) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = product) product
date_and_person_ assignment to item_definition_ instance_ relationship (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_definition_relationship) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = product_definition_relationship) (applied_date_and_organization_assignment

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = product_definition_relationship) product_definition_relationship {[(product_definition_relationship.related_product_definition -> product_definition => product_definition_occurrence) (product_definition_relationship => {(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')} product_definition_usage => assembly_component_usage)] [product_definition_relationship.relating_product_definition -> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template']}
date_and_person_ assignment to item_definition_ relationship (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_definition_relationship product_definition_relationship) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = product_definition_relationship product_definition_relationship) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				date_and_organization_item = product_definition_relationship product_definition_relationship) product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
date_and_person_ assignment to item_instance (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item (date_and_person_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (date_and_person_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] ->

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				date_and_person_and_organization_item (date_and_person_and_organization_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (date_and_person_and_organization_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item (date_and_organization_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (date_and_organization_item = assembly_component_usage

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')}))
date_and_person_ assignment to item_instance_ relationship (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item (date_and_person_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}) (date_and_person_item = product_definition_substitute product_definition_substitute)) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item (date_and_person_and_organization_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}) (date_and_person_and_organization_item = product_definition_substitute product_definition_substitute)) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item (date_and_organization_item = product_definition_relationship product_definition_relationship

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				{[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}) (date_and_organization_item = product_definition_substitute product_definition_substitute))
date_and_person_ assignment to item_solution (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = solution_based_product_definition) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = solution_based_product_definition) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = solution_based_product_definition) solution_based_product_definition
date_and_person_ assignment to item_version (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_definition_formation) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = product_definition_formation) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = product_definition_formation) product_definition_formation

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_and_person_ assignment to material (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = material_designation) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = material_designation) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = material_designation) material_designation
date_and_person_ assignment to physical_ instance (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_definition) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = product_definition) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = product_definition) product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_and_person_ assignment to physical_ instance_usage (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = assembly_component_usage) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = assembly_component_usage) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = assembly_component_usage) assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'physical occurrence usage'}
date_and_person_ assignment to process_plan (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = process_plan) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = process_plan) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = process_plan) process_plan
date_and_person_ assignment to product_class (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_class) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] ->

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				date_and_person_and_organization_item date_and_person_and_organization_item = product_class) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = product_class) product_class
date_and_person_ assignment to product_ component (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_component) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = product_component) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = product_component) product_component
date_and_person_ assignment to product_function (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_definition) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = product_definition) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = product_definition) product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				application_context_element.name = 'functional definition'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}
date_and_person_ assignment to product_ specification (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = configurable_item) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = configurable_item) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = configurable_item) configurable_item {configurable_item <= configuration_item configuration_item.description = 'product specification'}
date_and_person_ assignment to project (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = project) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = project) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				date_and_organization_item = project) project
date_and_person_ assignment to property (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item (date_and_person_item = action_property action_property) (date_and_person_item = property_definition property_definition)) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = property_definition) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item (date_and_organization_item = action_property action_property) (date_and_organization_item = property_definition property_definition))
date_and_person_ assignment to simulation_ information (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = kinematic_property_definition) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = kinematic_property_definition) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = kinematic_property_definition) kinematic_property_definition

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_and_person_ assignment to specification (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature))
date_and_person_ assignment to specification_ category (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category)) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = product_concept_feature_category (product_concept_feature_category)

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				(product_concept_feature_category => exclusive_product_concept_feature_category)) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] ->> date_and_organization_item date_and_organization_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category))
date_and_person_ assignment to specification_ expression (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] ->> date_and_person_item date_and_person_item = conditional_concept_feature) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] ->> date_and_person_and_organization_item date_and_person_and_organization_item = conditional_concept_feature) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] ->> date_and_organization_item date_and_organization_item = conditional_concept_feature) conditional_concept_feature {conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'expression'}
date_and_person_ assignment to specification_ inclusion (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] ->> date_and_person_item date_and_person_item = conditional_concept_feature) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] ->> date_and_person_and_organization_item date_and_person_and_organization_item = conditional_concept_feature) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] ->

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				date_and_organization_item date_and_organization_item = conditional_concept_feature) conditional_concept_feature {[conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'inclusion'] [conditional_concept_feature.condition -> concept_feature_relationship_with_condition concept_feature_relationship_with_condition.conditional_operator = \ 'implication']}
date_and_person_ assignment to styled_model (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item (date_and_person_item = draughting_model draughting_model) (date_and_person_item = \ mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item (date_and_person_and_organization_item = draughting_model draughting_model) (date_and_person_and_organization_item = \ mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item (date_and_organization_item = draughting_model draughting_model) (date_and_organization_item = \ mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation))

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_and_person_ assignment to view_area (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area))
date_and_person_ assignment to work_request (as is_applied_ to)	PATH			(applied_date_and_person_assignment applied_date_and_person_assignment.items[i] -> date_and_person_item date_and_person_item = versioned_action_request) (applied_date_and_person_and_organization_assignment applied_date_and_person_and_organization_assignment.items[i] -> date_and_person_and_organization_item date_and_person_and_organization_item = versioned_action_request) (applied_date_and_organization_assignment applied_date_and_organization_assignment.items[i] -> date_and_organization_item date_and_organization_item = versioned_action_request) versioned_action_request

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
DATE_AND_PERSON_ OR_ORGANIZATION	([(calendar_date) (date_and_time)] [(person) (person_and_ organization) (organization)]) (approval_person_ organization)	41 41 41 41 41 41		{approval_person_organization (date_and_time_item = approval_person_organization date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment) (date_item = approval_person_organization date_item <- applied_date_assignment.items[i] applied_date_assignment)}[1)]
date_and_person_ or_organization to date_time (as actual_date)	(IDENTICAL MAPPING) (PATH)			({(calendar_date <= date <- date_assignment.assigned_date date_assignment date_assignment.role -> date_role date_role.name = 'actual') (date_and_time <- date_and_time_assignment.assigned_date_and_time date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'actual')}) (approval_person_organization (date_item = approval_person_organization date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment date_assignment.assigned_date -> date => {date <- date_assignment.assigned_date date_assignment date_assignment.role ->

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				date_role date_role.name = 'actual'} calendar_date) (date_and_time_item = approval_person_organization date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.assigned_date_and_time -> date_and_time {date_and_time <- date_and_time_assignment.assigned_date_and_time date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'actual'}))
date_and_person_ or_organization to organization (as person_or_ organization)	(IDENTICAL MAPPING) (PATH)			(approval_person_organization approval_person_organization.person_organization -> person_organization_select person_organization_select = organization organization)
date_and_person_ or_organization to person (as person_or_ organization)	(IDENTICAL MAPPING) (PATH)			(approval_person_organization approval_person_organization.person_organization -> person_organization_select person_organization_select = person person)
date_and_person_ or_organization to person_in_ organization (as person_or_ organization)	(IDENTICAL MAPPING) (PATH)			(approval_person_organization approval_person_organization.person_organization -> person_organization_select person_organization_select = person_and_organization person_and_organization)
DATE_TIME	(calendar_date) (date_and_time)	41 41	22	

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date	(IDENTICAL MAPPING) (calendar_date)	 41		date_and_time date_and_time.date_component ->date => calendar_date
time	local_time	41		date_and_time date_and_time.time_component ->local_time
DATE_TIME_ASSIGNMENT	(applied_date_assignment) (applied_date_and_time_assignment)	214 214		(applied_date_assignment <= date_assignment) (applied_date_and_time_assignment <= date_and_time_assignment)
role	(date_role.name) (date_time_role.name)	41 41		(applied_date_assignment <= date_assignment date_assignment.role -> date_role date_role.name {(date_role.name) (date_role.name = 'creation') (date_role.name = 'installation') (date_role.name = 'production') (date_role.name = 'registration') (date_role.name = 'update')}) (applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name {(date_time_role.name) (date_time_role.name = 'creation') (date_time_role.name = 'installation') (date_time_role.name = 'production') (date_time_role.name = 'registration') (date_time_role.name = 'update')})

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_time_ assignment to date_time (as assigned_ date_time)	PATH			(applied_date_assignment <= date_assignment date_assignment.assigned_date -> date => calendar_date) (applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.assigned_date_and_time -> date_and_time)
date_time_ assignment to activity (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = action) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = action) action
date_time_ assignment to activity_element (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = applied_action_assignment) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = applied_action_assignment) applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_time_ assignment to assembly_ substitute_ relationship (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = assembly_component_usage_substitute) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = assembly_component_usage_substitute) assembly_component_usage_substitute
date_time_ assignment to class_category_ association (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_concept_feature_category_usage) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = product_concept_feature_category_usage) product_concept_feature_category_usage {product_concept_feature_category_usage <= group_assignment group_assignment.role -> group_role (group_role.name = 'mandatory category usage') (group_role.name = 'optional category usage')}
date_time_ assignment to class_condition_ association (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_concept_feature_association) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = product_concept_feature_association) product_concept_feature_association

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_time_ assignment to class_function_ association (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = configuration_design) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = configuration_design) configuration_design {configuration_design.configuration -> configuration_item configuration_item.description = 'class function definition'}
date_time_ assignment to class_inclusion_ association (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_concept_feature_association) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = product_concept_feature_association) product_concept_feature_association {product_concept_feature_association.name = 'inclusion association'}
date_time_ assignment to class_ specification_ association (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_concept_feature_association) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = product_concept_feature_association) product_concept_feature_association

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_time_ assignment to configuration (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item [date_item = configured_effective_item_assignment configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}] [date_item = configured_effective_context_assignment configured_effective_context_assignment {configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}]) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item [date_and_time_item = configured_effective_item_assignment configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}] [date_and_time_item = configured_effective_context_assignment configured_effective_context_assignment {configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}])

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_time_ assignment to design_ constraint (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = representation) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = representation) representation {representation.description = 'design constraint'}
date_time_ assignment to design_ discipline_item_ definition (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_definition) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = product_definition) product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
date_time_ assignment to drawing (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = auto_design_drawing_revision) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = auto_design_drawing_revision) auto_design_drawing_revision

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_time_ assignment to external_ document (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item (date_item = document document) (date_item = physically_modelled_product_definition physically_modelled_product_definition)) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item (date_and_time_item = document document) (date_and_time_item = physically_modelled_product_definition physically_modelled_product_definition))
date_time_ assignment to geometric_model (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = shape_representation) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = shape_representation) shape_representation
date_time_ assignment to item (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = product) product

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_time_ assignment to item_definition_ instance_ relationship (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_definition_relationship) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = product_definition_relationship) product_definition_relationship {[(product_definition_relationship.related_product_definition -> product_definition => product_definition_occurrence) (product_definition_relationship => {(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')} product_definition_usage => assembly_component_usage)] [product_definition_relationship.relating_product_definition -> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template']}
date_time_ assignment to item_definition_ relationship (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_definition_relationship) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = product_definition_relationship) product_definition_relationship {[product_definition_relationship.relating_product_definition ->]

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				[product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
date_time_ assignment to item_instance (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item (date_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (date_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item (date_and_time_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (date_and_time_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')}))
date_time_ assignment to item_instance_ relationship (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item (date_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}) (date_item = product_definition_substitute product_definition_substitute)) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item (date_and_time_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence})

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				(date_and_time_item = product_definition_substitute product_definition_substitute))
date_time_ assignment to item_solution (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = solution_based_product_definition) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = solution_based_product_definition) solution_based_product_definition
date_time_ assignment to item_version (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_definition_formation) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = product_definition_formation) product_definition_formation
date_time_ assignment to material (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = material_designation) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = material_designation) material_designation
date_time_ assignment to physical_ instance (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_definition) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				date_and_time_item = product_definition) product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}
date_time_ assignment to physical_ instance_usage (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = assembly_component_usage) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = assembly_component_usage) assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'physical occurrence usage'}
date_time_ assignment to process_plan (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = process_plan) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = process_plan) process_plan
date_time_ assignment to product_class (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_class) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				date_and_time_item = product_class) product_class
date_time_ assignment to product_ component (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_component) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = product_component) product_component
date_time_ assignment to product_function (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_definition) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = product_definition) product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_time_ assignment to product_ specification (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = configurable_item) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = configurable_item) configurable_item {configurable_item <= configuration_item configuration_item.description = 'product specification'}
date_time_ assignment to project (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = project) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = project) project
date_time_ assignment to property (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item (date_item = action_property action_property) (date_item = property_definition property_definition)) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item (date_and_time_item = action_property action_property) (date_and_time_item = property_definition property_definition))

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
date_time_ assignment to simulation_ information (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = kinematic_property_definition) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = kinematic_property_definition) kinematic_property_definition
date_time_ assignment to specification (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature))
date_time_ assignment to specification_ category (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category)) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category =>

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				exclusive_product_concept_feature_category))
date_time_ assignment to specification_ expression (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = conditional_concept_feature) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = conditional_concept_feature) conditional_concept_feature {conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'expression'}
date_time_ assignment to specification_ inclusion (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = conditional_concept_feature) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = conditional_concept_feature) conditional_concept_feature {[conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'inclusion'] [conditional_concept_feature.condition -> concept_feature_relationship_with_condition concept_feature_relationship_with_condition.conditional_operator = 'implication']}
date_time_ assignment to styled_model (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item (date_item = draughting_model draughting_model) (date_item = mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)) (applied_date_and_time_assignment

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_date_and_time_assignment.items[i] -> date_and_time_item (date_and_time_item = draughting_model draughting_model) (date_and_time_item = mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation))
date_time_ assignment to view_area (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area))
date_time_ assignment to work_request (as is_applied_ to)	PATH			(applied_date_assignment applied_date_assignment.items[i] -> date_item date_item = versioned_action_request) (applied_date_and_time_assignment applied_date_and_time_assignment.items[i] -> date_and_time_item date_and_time_item = versioned_action_request) versioned_action_request
DESIGN_ DISCIPLINE_ITEM_ DEFINITION	product_definition	41		{product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
id	product_definition. id	41		

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
name	product_definition. name	41		
design_ discipline_item_ definition to item_version (as associated_ item_version)	PATH			product_definition product_definition.formation ->
product_definition_formation
design_ discipline_item_ definition to application_ context (as is_relevant_ for)	PATH			product_definition product_definition.frame_of_reference ->
(product_definition_context)
(product_definition_context <=
application_context_element <-
(application_context_element_relationship.related_context_element)
(application_context_element_relationship.relating_context_element)
application_context_element_relationship
{application_context_element_relationship *}
(application_context_element_relationship.related_context_element ->)
(application_context_element_relationship.relating_context_element ->)
application_context_element =>
product_definition_context)
design_ discipline_item_ definition to multi_language_ string (as name)	PATH		140	product_definition multi_language_attribute_item = product_definition
multi_language_attribute_item <-
applied_multi_language_attribute_assignment.items[1]
applied_multi_language_attribute_assignment
{applied_multi_language_attribute_assignment <=
multi_language_attribute_assignment
multi_language_attribute_assignment.attribute_name = 'name'}
ITEM	product	41	70	{product product.frame_of_reference[i] ->
product_context <=
application_context_element
application_context_element.frame_of_reference[i] ->
application_context <-
application_protocol_definition.application[i]

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				application_protocol_definition application_protocol_definition.application_interpreted_model_schema_name = \ 'automotive_design'}
description	product.description	41		
id	product.id	41		
name	product.name	41		
item to multi_ language_string (as description)	PATH		133	product multi_language_attribute_item = product multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'description'}
item to multi_ language_string (as name)	PATH		133	product multi_language_attribute_item = product multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'name'}
ITEM_ CLASSIFICATION	product_category	41		
description	product_category. description	41		
item_ classification to item (as associated_ item)	PATH			product_category => product_related_product_category product_related_product_category.products[i] ->\ product

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
item_ classification to multi_ language_string (as description)	PATH		134	product_category multi_language_attribute_item = product_category multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
ITEM_VERSION	product_definition_ formation	41		
description	product_definition_ formation. description	41		
id	product_definition_ formation.id	41		
item_version to item (as associated_ item)	PATH			product_definition_formation product_definition_formation.of_product -> product
item_version to multi_language_ string (as description)	PATH		141	product_definition_formation multi_language_attribute_item = product_definition_formation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
MULTI_LANGUAGE_ STRING	applied_multi_ language_attribute_ assignment	214		applied_multi_language_attribute_assignment <= multi_language_attribute_assignment

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
multi_language_ string to string_ with_language (as language_ dependent_ string)	PATH			applied_multi_language_attribute_assignment <= multi_language_attribute_assignment
ORGANIZATION	organization	41		
delivery_address	organizational_ address	41		organization <- organizational_address.organizations[i] organizational_address {organizational_address.description = 'delivery address'}
id	organization.id	41		
name	organization.name	41		
organization_ type	organization. description	41		
postal_address	organizational_ address	41		organization <- organizational_address.organizations[i] organizational_address {organizational_address.description = 'postal address'}
visitor_address	organizational_ address	41		organization <- organizational_address.organizations[i] organizational_address {organizational_address.description = 'visitor address'}
PERSON	person	41	66	{person <- person_and_organization.the_person person_and_organization}
address	personal_address	41		person <- personal_address.people[i] personal_address
id	person.id	41		
name	<person.last_name> <person.first_name> <person.middle_ names>	41 41 41		

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
PERSON_IN_ ORGANIZATION	person_and_ organization	41		
role	person_and_ organization.name	41		
person_in_ organization to organization (as organization)	PATH			person_and_organization person_and_organization.the_organization ->organization
person_in_ organization to person (as person)	PATH			person_and_organization person_and_organization.the_person ->person
PERSON_OR_ ORGANIZATION_ ASSIGNMENT	(applied_person_ assignment) (applied_person_and_ organization_ assignment) (applied_ organization_ assignment)	214 214 214		(applied_person_assignment <= person_assignment) (applied_person_and_organization_assignment <= person_and_organization_assignment) (applied_organization_assignment <= organization_assignment)
role	(person_role.name) (person_and_ organization_role. name) (organization_role. name)	41 41 41	39,37	(applied_person_assignment <= person_assignment person_assignment.role -> person_role person_role.name {(person_role.name) (person_role.name = 'creator') (person_role.name = 'editor') (person_role.name = 'owner') (person_role.name = 'customer') (person_role.name = 'custodian')}) (applied_person_and_organization_assignment <= person_and_organization_assignment person_and_organization_assignment.role ->

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				person_and_organization_role person_and_organization_role.name {(person_and_organization_role.name) (person_and_organization_role.name = 'creator') (person_and_organization_role.name = 'editor') (person_and_organization_role.name = 'owner') (person_and_organization_role.name = 'customer') (person_and_organization_role.name = 'custodian')}) (applied_organization_assignment <= organization_assignment organization_assignment.role -> organization_role organization_role.name {(organization_role.name) (organization_role.name = 'creator') (organization_role.name = 'editor') (organization_role.name = 'owner') (organization_role.name = 'customer') (organization_role.name = 'custodian')})
person_or_ organization_ assignment to organization (as assigned_ person_or_ organization)	PATH			applied_organization_assignment <= organization_assignment organization_assignment.assigned_organization -> organization
person_or_ organization_ assignment to person (as assigned_ person_or_ organization)	PATH			applied_person_assignment <= person_assignment person_assignment.assigned_person -> person

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
person_or_ organization_ assignment to person_in_ organization (as assigned_ person_or_ organization)	PATH			applied_person_and_organization_assignment <= person_and_organization_assignment person_and_organization_assignment.assigned_person_and_organization -> person_and_organization
person_or_ organization_ assignment to activity (as is_applied_ to)	PATH		3	(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = action) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = action) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = action) action
person_or_ organization_ assignment to activity_element (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = applied_action_assignment) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = applied_action_assignment) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = applied_action_assignment) applied_action_assignment {applied_action_assignment <= action_assignment

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_assignment.role -> action_role action_role.description = 'activity element'}
person_or_ organization_ assignment to assembly_ substitute_ relationship (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = assembly_component_usage_substitute) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = assembly_component_usage_substitute) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = assembly_component_usage_substitute) assembly_component_usage_substitute
person_or_ organization_ assignment to class_category_ association (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_concept_feature_category_usage) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_concept_feature_category_usage) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product_concept_feature_category_usage) product_concept_feature_category_usage {product_concept_feature_category_usage <= group_assignment group_assignment.role -> group_role (group_role.name = 'mandatory category usage') (group_role.name = 'optional category usage')}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
person_or_ organization_ assignment to class_condition_ association (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_concept_feature_association) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_concept_feature_association) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product_concept_feature_association) product_concept_feature_association
person_or_ organization_ assignment to class_function_ association (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = configuration_design) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = configuration_design) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = configuration_design) configuration_design {configuration_design.configuration -> configuration_item configuration_item.description = 'class function definition'}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
person_or_ organization_ assignment to class_inclusion_ association (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_concept_feature_association) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_concept_feature_association) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product_concept_feature_association) product_concept_feature_association {product_concept_feature_association.name = 'inclusion association'}
person_or_ organization_ assignment to class_ specification_ association (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_concept_feature_association) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_concept_feature_association) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product_concept_feature_association) product_concept_feature_association
person_or_ organization_ assignment to configuration (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item [person_item = configured_effective_item_assignment configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				effectivity_role.name = 'configuration'}] [person_item = configured_effective_context_assignment configured_effective_context_assignment {configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}]) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item [person_and_organization_item = configured_effective_item_assignment configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}] [person_and_organization_item = configured_effective_context_assignment configured_effective_context_assignment {configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}]) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item [organization_item = configured_effective_item_assignment configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				effectivity_role.name = 'configuration'}] [organization_item = configured_effective_context_assignment configured_effective_context_assignment {configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}])
person_or_ organization_ assignment to design_ constraint (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = representation) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = representation) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = representation) representation {representation.description = 'design constraint'}
person_or_ organization_ assignment to design_ discipline_item_ definition (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_definition) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_definition) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product_definition) product_definition {product_definition.frame_of_reference ->

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
person_or_ organization_ assignment to drawing (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = auto_design_drawing_revision) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = auto_design_drawing_revision) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = auto_design_drawing_revision) auto_design_drawing_revision
person_or_ organization_ assignment to external_ document (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item (person_item = document document) (person_item = physically_modelled_product_definition physically_modelled_product_definition)) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item (person_and_organization_item = document document) (person_and_organization_item = physically_modelled_product_definition physically_modelled_product_definition)) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item (organization_item = document

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				document) (organization_item = physically_modelled_product_definition physically_modelled_product_definition))
person_or_ organization_ assignment to geometric_model (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = shape_representation) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item =shape_representation) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = shape_representation) shape_representation
person_or_ organization_ assignment to item (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product) product

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
person_or_ organization_ assignment to item_definition_ instance_ relationship (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_definition_relationship) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_definition_relationship) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product_definition_relationship) product_definition_relationship {[(product_definition_relationship.related_product_definition -> product_definition => product_definition_occurrence) (product_definition_relationship => {(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')} product_definition_usage => assembly_component_usage)] [product_definition_relationship.relating_product_definition -> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template']}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
person_or_ organization_ assignment to item_definition_ relationship (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_definition_relationship) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_definition_relationship) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product_definition_relationship) product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
person_or_ organization_ assignment to item_instance (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item (person_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (person_item = assembly_component_usage assembly_component_usage

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				{assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item (person_and_organization_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (person_and_organization_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item (organization_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <=

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition [product_definition.frame_of_reference ->] product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (organization_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')}))
person_or_organization_assignment to item_instance_relationship (as is_applied_to)	PATH			(applied_person_assignment applied_person_assignment.items[i] ->] person_item (person_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}) (person_item = product_definition_substitute product_definition_substitute)) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] ->] person_and_organization_item (person_and_organization_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition =>

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_occurrence}) (person_and_organization_item = product_definition_substitute product_definition_substitute)) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item (organization_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}) (organization_item = product_definition_substitute product_definition_substitute))
person_or_ organization_ assignment to item_solution (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = solution_based_product_definition) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = solution_based_product_definition) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = solution_based_product_definition) solution_based_product_definition
person_or_ organization_ assignment to item_version (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_definition_formation) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_definition_formation)

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				(applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product_definition_formation) product_definition_formation
person_or_ organization_ assignment to material (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = material_designation) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = material_designation) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = material_designation) material_designation
person_or_ organization_ assignment to physical_ instance (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_definition) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_definition) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product_definition) product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
person_or_ organization_ assignment to physical_ instance_usage (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = assembly_component_usage) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = assembly_component_usage) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = assembly_component_usage) assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'physical occurrence usage'}
person_or_ organization_ assignment to process_plan (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = process_plan) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = process_plan) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = process_plan) process_plan

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
person_or_ organization_ assignment to product_class (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_class) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_class) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product_class) product_class
person_or_ organization_ assignment to product_ component (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_component) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_component) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product_component) product_component
person_or_ organization_ assignment to product_function (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_definition) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_definition) (applied_organization_assignment applied_organization_assignment.items[i] ->

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				organization_item organization_item = product_definition) product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}
person_or_ organization_ assignment to product_ specification (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = configurable_item) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = configurable_item) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = configurable_item) configurable_item {configurable_item <= configuration_item configuration_item.description = 'product specification'}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
person_or_ organization_ assignment to project (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = project) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = project) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = project) project
person_or_ organization_ assignment to property (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item (person_item = action_property action_property) (person_item = property_definition property_definition)) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item (person_and_organization_item = action_property action_property) (person_and_organization_item = property_definition property_definition)) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item (organization_item = action_property action_property) (organization_item = property_definition property_definition))

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
person_or_ organization_ assignment to simulation_ information (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = kinematic_property_definition) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = kinematic_property_definition) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = kinematic_property_definition) kinematic_property_definition
person_or_ organization_ assignment to specification (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature))

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
person_or_ organization_ assignment to specification_ category (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category)) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category)) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category))
person_or_ organization_ assignment to specification_ expression (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = conditional_concept_feature) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = conditional_concept_feature) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = conditional_concept_feature) conditional_concept_feature

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				{conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'expression'}
person_or_ organization_ assignment to specification_ inclusion (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = conditional_concept_feature) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = conditional_concept_feature) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = conditional_concept_feature) conditional_concept_feature {[conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'inclusion'] [conditional_concept_feature.condition -> concept_feature_relationship_with_condition concept_feature_relationship_with_condition.conditional_operator = \ 'implication']}
person_or_ organization_ assignment to styled_model (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item (person_item = draughting_model draughting_model) (person_item = mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item (person_and_organization_item = draughting_model draughting_model)

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				(person_and_organization_item = \ mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item (organization_item = draughting_model draughting_model) (organization_item = \ mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation))
person_or_ organization_ assignment to view_area (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area))

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
person_or_ organization_ assignment to work_request (as is_applied_ to)	PATH			(applied_person_assignment applied_person_assignment.items[i] -> person_item person_item = versioned_action_request) (applied_person_and_organization_assignment applied_person_and_organization_assignment.items[i] -> person_and_organization_item person_and_organization_item = versioned_action_request) (applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = versioned_action_request) versioned_action_request
SECURITY_ CLASSIFICATION	security_ classification	41	184	
name	security_ classification.name	41		
purpose	security_ classification. purpose	41		
security_ classification to activity (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = action action
security_ classification to activity_ element (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = applied_action_assignment applied_action_assignment

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				{applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'}
security_ classification to assembly_ substitute_ relationship (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = assembly_component_usage_substitute assembly_component_usage_substitute
security_ classification to class_ function_ association (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = configuration_design configuration_design {configuration_design.configuration -> configuration_item configuration_item.description = 'class function definition'}
security_ classification to configuration (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item [security_classification_item = configured_effective_item_assignment configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				effectivity_role.name = 'configuration'}] [security_classification_item = configured_effective_context_assignment configured_effective_context_assignment {configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}]
security_ classification to design_ constraint (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = representation representation {representation.description = 'design constraint'}
security_ classification to drawing (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = auto_design_drawing_revision auto_design_drawing_revision
security_ classification to external_ document (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item (security_classification_item = document document) (security_classification_item = physically_modelled_product_definition physically_modelled_product_definition)

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
security_ classification to geometric_ model (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = shape_representation shape_representation
security_ classification to item (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = product product
security_ classification to item_solution (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = solution_based_product_definition solution_based_product_definition
security_ classification to item_version (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = product_definition_formation product_definition_formation

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
security_ classification to material (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = material_designation material_designation
security_ classification to physical_ instance (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}
security_ classification to physical_ instance_usage (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'physical occurrence usage'}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
security_ classification to process_plan (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = process_plan process_plan
security_ classification to product_class (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = product_class product_class
security_ classification to product_ component (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = product_component product_component
security_ classification to product_ function (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition']}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
				[product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}
security_ classification to project (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = project project
security_ classification to property (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item (security_classification_item = property_definition property_definition) (security_classification_item = action_property action_property)
security_ classification to simulation_ information (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = kinematic_property_definition kinematic_property_definition

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
security_ classification to specification (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)
security_ classification to specification_ category (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category)
security_ classification to styled_model (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item (security_classification_item = draughting_model draughting_model) (security_classification_item = \ mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
security_ classification to view_area (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)
security_ classification to work_order (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = action_directive action_directive
security_ classification to work_request (as is_applied_ to)	PATH		44	security_classification <- security_classification_assignment.assigned_security_classification security_classification_assignment => applied_security_classification_assignment applied_security_classification_assignment.items[i] -> security_classification_item security_classification_item = versioned_action_request versioned_action_request
security_ classification to multi_ language_string (as purpose)	PATH		158	security_classification multi_language_attribute_item = security_classification multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'purpose'}

Table 23 – Mapping table product_management_data UoF (S1) (continued)

Application element	AIM element	Source	Rules	Reference path
security_ classification to security_ level (as security_ classification_ level)	PATH			security_classification security_classification.security_level -> security_classification_level
SECURITY_LEVEL	security_ classification_level	41		
name	security_ classification_level. name	41		
security_level to classification_ system (as used_ classification_ system)	PATH			security_classification_level <- group_member_item = security_classification_level group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment {group_assignment.role -> group_role group_role.name = 'classified member'} group_assignment.assigned_group -> group => (classification_system) (externally_defined_classification_system)
SPECIFIC_ CLASSIFICATION	product_category	41		
classification_ name	product_category. name	41		{product_category [(product_category.name) (product_category.name = 'part') (product_category.name = 'raw material') (product_category.name = 'tool')] [product_category <- product_category_relationship.sub_category]}

Table 23 – Mapping table product_management_data UoF (S1) (concluded)

Application element	AIM element	Source	Rules	Reference path
classification_ type	product_category. name	41		product_category <- {(product_category.name) (product_category.name = 'part') (product_category.name = 'raw material') (product_category.name = 'tool')} product_category_relationship.sub_category product_category_relationship product_category_relationship.category -> product_category product_category.name {product_category.name = 'item role'}
STRING_WITH_ LANGUAGE	multi_language_ attribute_assignment	41		
contents	multi_language_ attribute_assignment. alternate_attribute_ value	41		
language	multi_language_ attribute_assignment. language	41		
primary	language_role.name	41	34	multi_language_attribute_assignment multi_language_attribute_assignment.role -> language_role language_role.name {(language_role.name = 'primary') (language_role.name = 'translated')}

NOTE – The following note numbers correspond to the footnotemarks listed in the table above:

1 approval_person_organization is only to be used for the ARM assertion 'approval to date_and_-person_or_organization (as is_approved_by).

Table 24 – Mapping table element_structure UoF (S2)

Application element	AIM element	Source	Rules	Reference path
ACCURACY	(global_uncertainty_ assigned_context) (uncertainty_ assigned_ representation) (qualified_ representation_item)	43 43 45		
angular_accuracy_ value	(uncertainty_measure_ with_unit) (qualitative_ uncertainty)	43 45	49	(global_uncertainty_assigned_context global_uncertainty_assigned_context.uncertainty[i] -> uncertainty_measure_with_unit {uncertainty_measure_with_unit.name = 'angular accuracy value'} {uncertainty_measure_with_unit <= measure_with_unit => plane_angle_measure_with_unit}) (uncertainty_assigned_representation uncertainty_assigned_representation.uncertainty[i] -> uncertainty_measure_with_unit {uncertainty_measure_with_unit.name = 'angular accuracy value'} {uncertainty_measure_with_unit <= measure_with_unit => plane_angle_measure_with_unit}) (qualified_representation_item qualified_representation_item.qualifiers[i] -> value_qualifier = uncertainty_qualifier uncertainty_qualifier => {uncertainty_qualifier.measure_name ='angular accuracy value' } qualitative_uncertainty)
curvature_ accuracy_value	(uncertainty_measure_ with_unit) (qualitative_ uncertainty)	43 45	49	(global_uncertainty_assigned_context global_uncertainty_assigned_context.uncertainty[i] -> uncertainty_measure_with_unit {[uncertainty_measure_with_unit.name = 'curvature accuracy value'] [uncertainty_measure_with_unit <= measure_with_unit => ratio_measure_with_unit]})

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
				(uncertainty_assigned_representation uncertainty_assigned_representation.uncertainty[i] -> uncertainty_measure_with_unit {[uncertainty_measure_with_unit.name = 'curvature accuracy value'] [uncertainty_measure_with_unit <= measure_with_unit => ratio_measure_with_unit]}) (qualified_representation_item qualified_representation_item.qualifiers[i] -> value_qualifier = uncertainty_qualifier uncertainty_qualifier => {uncertainty_qualifier.measure_name ='curvature accuracy value' } qualitative_uncertainty)
distance_ accuracy_value	(uncertainty_measure_ with_unit) (qualitative_ uncertainty)	43 45	49	(global_uncertainty_assigned_context global_uncertainty_assigned_context.uncertainty[i] -> uncertainty_measure_with_unit {uncertainty_measure_with_unit.name = 'distance accuracy value'} {uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit}) (uncertainty_assigned_representation uncertainty_assigned_representation.uncertainty[i] -> uncertainty_measure_with_unit {uncertainty_measure_with_unit.name = 'distance accuracy value'} {uncertainty_measure_with_unit <= measure_with_unit => length_measure_with_unit}) (qualified_representation_item qualified_representation_item.qualifiers[i] -> value_qualifier = uncertainty_qualifier uncertainty_qualifier => {uncertainty_qualifier.measure_name ='distance accuracy value' }

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
				qualitative_uncertainty)
accuracy to face (as is_defined_ for)	PATH			qualified_representation_item <= representation_item => geometric_representation_item => face_surface => advanced_face
accuracy to geometric_model (as is_defined_ for)	PATH			(global_uncertainty_assigned_context <= representation_context <- representation.context_of_items representation => shape_representation) (uncertainty_assigned_representation <= representation => shape_representation)
accuracy to oriented_edge_ curve (as is_defined_ for)	PATH			qualified_representation_item <= representation_item => topological_representation_item => edge => oriented_edge
accuracy to vertex_point (as is_defined_ for)	PATH			qualified_representation_item <= representation_item => geometric_representation_item => vertex_point
AXIS_PLACEMENT	placement	42		{placement => (axis1_placement) (axis2_placement_2d) (axis2_placement_3d)}
CARTESIAN_ COORDINATE_SPACE	geometric_ representation_ context	42	50	
unit	global_unit_assigned_ context.units	41	62	geometric_representation_context <= representation_context => global_unit_assigned_context global_unit_assigned_context.units

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
CARTESIAN_ COORDINATE_SPACE_ 2D	geometric_ representation_ context	42		{geometric_representation_context geometric_representation_context.coordinate_space_dimension = 2}
CARTESIAN_ COORDINATE_SPACE_ 3D	geometric_ representation_ context	42		{geometric_representation_context geometric_representation_context.coordinate_space_dimension = 3}
GEOMETRIC_MODEL_ RELATIONSHIP_ WITH_ TRANSFORMATION	representation_ relationship_with_ transformation	43		
geometric_model_ relationship_ with_ transformation to transformation (as transformation)	PATH			representation_relationship_with_transformation representation_relationship_with_transformation.transformation_operator -> transformation (transformation = item_defined_transformation item_defined_transformation {([item_defined_transformation.transform_item_1 -> representation_item => geometric_representation_item => placement => axis2_placement_2d] [item_defined_transformation.transform_item_2 -> representation_item => geometric_representation_item => (placement => axis2_placement_2d) (cartesian_transformation_operator => {cartesian_transformation_operator.scl = 1.0} cartesian_transformation_operator_2d)]) ([item_defined_transformation.transform_item_1 -> representation_item => geometric_representation_item => placement => axis2_placement_3d] [item_defined_transformation.transform_item_2 -> representation_item => geometric_representation_item =>

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
				(placement => axis2_placement_3d) (cartesian_transformation_operator => {cartesian_transformation_operator.scl = 1.0} cartesian_transformation_operator_3d)])}) (transformation = functionally_defined_transformation functionally_defined_transformation {functionally_defined_transformation => cartesian_transformation_operator => {cartesian_transformation_operator.scl = 1.0} (cartesian_transformation_operator_2d) (cartesian_transformation_operator_3d)})
GROUP	(group) (rep_item_group)	41 214		rep_item_group <= [group] [representation_item]
name	group.name	41		
purpose	group.description	41		
group to detailed_element (as element)	PATH			(group <-) (rep_item_group <= group <-) group_assignment.assigned_group group_assignment => group_member_assignment group_member_assignment.items[i] -> group_member_item = representation_item representation_item
group to multi_ language_string (as name)	PATH		116,152	(group multi_language_attribute_item = group) (rep_item_group multi_language_attribute_item = rep_item_group) multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment ({applied_multi_language_attribute_assignment <= multi_language_attribute_assignment

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
				multi_language_attribute_assignment.attribute_name = 'name'}) ({applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'group.name'})
group to multi_language_string (as purpose)	PATH		116,152	(group multi_language_attribute_item = group) (rep_item_group multi_language_attribute_item = rep_item_group) multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment ({applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}) ({applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'group.description'})
GROUP_RELATIONSHIP	group_relationship	41		
description	group_relationship.description	41		
relation_type	group_relationship.name	41		
group_relationship to multi_language_string (as description)	PATH			group_relationship multi_language_attribute_item = group_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
group_relationship to group (as related)	PATH			group_relationship group_relationship.related_group ->(group) (group => rep_item_group)

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
group_ relationship to group (as relating)	PATH			group_relationship group_relationship.relating_group ->(group) (group =>rep_item_group)
ITEM_SHAPE	product_definition_ shape	41		
description	property_definition. description	41		product_definition_shape <= property_definition property_definition.description
item_shape to design_ discipline_item_ definition (as described_ object)	PATH			product_definition_shape <= property_definition property_definition.definition ->characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition
item_shape to item_definition_ instance_ relationship (as described_ object)	PATH			product_definition_shape <= property_definition property_definition.definition ->characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship
item_shape to item_definition_ relationship (as described_ object)	PATH			product_definition_shape <= property_definition property_definition.definition ->characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship
item_shape to item_instance (as described_ object)	PATH			product_definition_shape <= property_definition property_definition.definition ->characterized_definition

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition => product_definition_occurrence) (characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship => product_definition_usage => assembly_component_usage)
item_shape to item_instance_ relationship (as described_ object)	PATH			product_definition_shape <= property_definition property_definition.definition ->characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}
item_shape to multi_language_ string (as description)	PATH		145	product_definition_shape multi_language_attribute_item = product_definition_shape multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'property_definition.description'}
LAYER	presentation_layer_ assignment	46		
description	presentation_layer_ assignment. description	46		

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
id	presentation_layer_ assignment.name	46		
layer to multi_ language_string (as description)	PATH		127	presentation_layer_assignment multi_language_attribute_item = presentation_layer_assignment multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'description'}
layer to detailed_element (as element)	PATH		67,52	presentation_layer_assignment presentation_layer_assignment.assigned_items[i] -> layered_item = representation_item representation_item
MODEL_CHANGE	(model_change) (process_change)	214 214		(model_change <= property_definition) (process_change <= action_property)
description	(property_definition. description) (action_property. description)	41 49		(model_change <= property_definition property_definition.description) (process_change <= action_property action_property.description)
id	(property_definition. name) (action_property. name)	41 49		(model_change <= property_definition property_definition.name) (process_change <= action_property action_property.name)
model_change to detailed_element (as added_ element)	PATH			(model_change <= property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation {property_definition_representation.name = 'added elements'} property_definition_representation.used_representation ->) (process_change <= action_property <- action_property_representation.property action_property_representation {action_property_representation.name = 'added elements'} action_property_representation.representation ->) representation representation.items[i] -> representation_item {representation_item <- representation.items[i] representation <- representation_relationship.rep_2 representation_relationship representation_relationship.rep_1 -> representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.name = 'relating change_description'}
model_change to geometric_model_ relationship (as change_ description)	PATH			(model_change <= property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation.name = 'relating change_description'} property_definition_representation.used_representation ->) (process_change <= action_property <- action_property_representation.property

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_property_representation {action_property_representation.name = 'relating change_description'} action_property_representation.representation ->) representation <- representation_relationship.rep_1 representation_relationship {representation_relationship.rep_2 -> representation <- (property_definition_representation.used_representation property_definition_representation {property_definition_representation.name = 'related change_description'} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition => model_change) (action_property_representation.representation action_property_representation {action_property_representation.name = 'related change_description'} action_property_representation.property -> action_property => process_change)}
model_change to styled_model_ relationship (as change_ description)	PATH			(model_change <= property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation.name = 'relating change_description'} property_definition_representation.used_representation ->) (process_change <= action_property <-

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_property_representation.property action_property_representation {action_property_representation.name = 'relating change_description'} action_property_representation.representation ->) representation <- representation_relationship.rep_1 representation_relationship {representation_relationship.rep_2 -> representation <- (property_definition_representation.used_representation property_definition_representation {property_definition_representation.name = 'related change_description'} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition => model_change) (action_property_representation.representation action_property_representation {action_property_representation.name = 'related change_description'} action_property_representation.property -> action_property => process_change)}
model_change to view_area_ relationship (as change_ description)	PATH			(model_change <= property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation.name = 'relating change_description'} property_definition_representation.used_representation ->) (process_change <=

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_property <- action_property_representation.property action_property_representation {action_property_representation.name = 'relating change_description'} action_property_representation.representation ->) representation <- representation_relationship.rep_1 representation_relationship => {representation_relationship.rep_2 -> representation <- (property_definition_representation.used_representation property_definition_representation {property_definition_representation.name = 'related change_description'} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition => model_change) (action_property_representation.representation action_property_representation {action_property_representation.name = 'related change_description'} action_property_representation.property -> action_property => process_change)} area_presentation_relationship
model_change to detailed_element (as deleted_ element)	PATH			(model_change <= property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation.name = \ 'deleted elements'}

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_representation ->) (process_change <= action_property <- action_property_representation.property action_property_representation {action_property_representation.name = 'deleted elements'} action_property_representation.representation ->) representation representation.items[i] -> representation_item {representation_item <- representation.items[i] representation <- representation_relationship.rep_2 representation_relationship representation_relationship.rep_1 -> representation <- property_definition_representation.used_representation property_definition_representation property_definition_representation.name = 'related change_description'}
model_change to multi_language_ string (as description)	PATH		122,130	(model_change multi_language_attribute_item = model_change multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'property_definition.description'}) (process_change multi_language_attribute_item = process_change multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1]

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'action_property.description'})
model_change to change (as is_describing)	PATH			(model_change <= property_definition property_definition.definition -> characterized_definition (characterized_definition = characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship) (characterized_definition = shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship)) (process_change <= action_property action_property.definition -> characterized_action_definition = action_relationship action_relationship)
SHAPE_ASPECT	shape_aspect	41		
description	shape_aspect. description	41		
shape_aspect to item_shape (as composition)	PATH			shape_aspect shape_aspect.of_shape -> product_definition_shape
shape_aspect to multi_language_string (as description)	PATH		159	shape_aspect multi_language_attribute_item = shape_aspect multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
SHAPE_ASPECT_ RELATIONSHIP	shape_aspect_ relationship	41		
description	shape_aspect_ relationship. description	41		
relation_type	shape_aspect_ relationship.name	41		
shape_aspect_ relationship to multi_language_ string (as description)	PATH		160	shape_aspect_relationship multi_language_attribute_item = shape_aspect_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
shape_aspect_ relationship to shape_aspect (as related)	PATH			shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect
shape_aspect_ relationship to shape_aspect (as relating)	PATH			shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect -> shape_aspect
SHAPE_ DESCRIPTION_ ASSOCIATION	(shape_definition_ representation) (property_definition_ representation)	41 41		
shape_ description_ association to external_model (as defining_ geometry)	PATH			property_definition_representation property_definition_representation.used_representation -> representation => externally_defined_representation

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
shape_ description_ association to geometric_model (as defining_ geometry)	PATH			shape_definition_representation <= property_definition_representation property_definition_representation.used_representation -> representation => shape_representation
shape_ description_ association to feature_ definition (as is_defining_ shape_for)	PATH			shape_definition_representation <= property_definition_representation property_definition_representation.definition -> represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition = characterized_object characterized_object => feature_definition
shape_ description_ association to general_feature (as is_defining_ shape_for)	PATH			shape_definition_representation <= property_definition_representation property_definition_representation.definition -> represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect
shape_ description_ association to item_shape (as is_defining_ shape_for)	PATH			shape_definition_representation <= property_definition_representation property_definition_representation.definition -> represented_definition = property_definition property_definition => product_definition_shape

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
shape_ description_ association to shape_aspect (as is_defining_ shape_for)	PATH			shape_definition_representation <= property_definition_representation property_definition_representation.definition -> represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect
shape_ description_ association to shape_aspect_ relationship (as is_defining_ shape_for)	PATH			shape_definition_representation <= property_definition_representation property_definition_representation.definition -> represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition = shape_definition shape_definition = shape_aspect_relationship shape_aspect_relationship
shape_ description_ association to thread_feature (as is_defining_ shape_for)	PATH			shape_definition_representation <= property_definition_representation property_definition_representation.definition -> represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect => thread_occurrence

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
shape_ description_ association to transition_ feature (as is_defining_ shape_for)	PATH			shape_definition_representation <= property_definition_representation property_definition_representation.definition -> represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition = shape_definition shape_definition = shape_aspect shape_aspect => transition_feature
STRUCTURING_ METHOD_HIERARCHY	(group_relationship) (presentation_layer_ assignment)	41 46		{group_relationship.name = 'hierarchy'}
structuring_ method_hierarchy to group (as related)	PATH			(group_relationship group_relationship.related_group -> (group => rep_item_group) (group)) (presentation_layer_assignment presentation_layer_assignment.assigned_items[i] -> layered_item = representation_item representation_item => rep_item_group)
structuring_ method_hierarchy to group (as relating)	PATH			group_relationship group_relationship.relating_group -> (group => rep_item_group) (group)
structuring_ method_hierarchy to layer (as relating)	IDENTICAL MAPPING			{presentation_layer_assignment.assigned_items[i] -> representation_item => rep_item_group}

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
TEMPLATE_ INSTANCE	mapped_item	43		
id	representation_item. name	43		mapped_item <= representation_item representation_item.name
scale	cartesian_ transformation_ operator.scale	42		mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => cartesian_transformation_operator cartesian_transformation_operator.scale {cartesian_transformation_operator => (cartesian_transformation_operator_2d) (cartesian_transformation_operator_3d)}
template_ instance to external_model (as template_ definition)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => externally_defined_representation
template_ instance to geometric_model (as template_ definition)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => shape_representation
template_ instance to styled_model (as template_ definition)	PATH			mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation -> representation => (draughting_model) (mechanical_design_geometric_presentation_representation)

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
template_ instance to transformation (as transformation)	PATH	43		mapped_item [mapped_item.mapping_target] [mapped_item.mapping_source -> representation_map representation_map.mapping_origin]
TRANSFORMATION	([mapped_item. mapping_target] [representation_map. mapping_origin]) (item_defined_ transformation) (functionally_ defined_ transformation)	43 43 43 43		
TRANSFORMATION_ 2D	([mapped_item. mapping_target] [representation_map. mapping_origin]) (item_defined_ transformation) (functionally_ defined_ transformation)	43 43 43 43		{([mapped_item.mapping_target -> representation_item => geometric_representation_item => (placement => axis2_placement_2d) (cartesian_transformation_operator => cartesian_transformation_operator_2d)] [representation_map.mapping_origin -> representation_item => geometric_representation_item => placement => axis2_placement_2d]) (item_defined_transformation [item_defined_transformation.transform_item_1 -> representation_item => geometric_representation_item => placement => axis2_placement_2d] [item_defined_transformation.transform_item_2 -> representation_item => geometric_representation_item =>

Table 24 – Mapping table element_structure UoF (S2) (continued)

Application element	AIM element	Source	Rules	Reference path
				(placement => axis2_placement_2d) (cartesian_transformation_operator => cartesian_transformation_operator_2d)]) (functionally_defined_transformation => cartesian_transformation_operator => cartesian_transformation_operator_2d)}
TRANSFORMATION_ 3D	([mapped_item. mapping_target] [representation_map. mapping_origin]) (item_defined_ transformation) (functionally_ defined_ transformation)	43 43 43 43		{([mapped_item.mapping_target -> representation_item => geometric_representation_item => (placement => axis2_placement_3d) (cartesian_transformation_operator => cartesian_transformation_operator_3d)] [representation_map.mapping_origin -> representation_item => geometric_representation_item => placement => axis2_placement_3d]) (item_defined_transformation [item_defined_transformation.transform_item_1 -> representation_item => geometric_representation_item => placement => axis2_placement_3d] [item_defined_transformation.transform_item_2 -> representation_item => geometric_representation_item => (placement => axis2_placement_3d) (cartesian_transformation_operator => cartesian_transformation_operator_3d)]) (functionally_defined_transformation =>

Table 24 – Mapping table element_structure UoF (S2) (concluded)

Application element	AIM element	Source	Rules	Reference path
				cartesian_transformation_operator => cartesian_transformation_operator_3d)}

Table 25 – Mapping table item_definition_structure UoF (S3)

Application element	AIM element	Source	Rules	Reference path
ASSEMBLY_ COMPONENT_ RELATIONSHIP	assembly_component_ usage	44		{assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relating_product_definition -> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
assembly_ component_ relationship to assembly_ definition (as relating)	PATH			assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.relating_product_definition -> product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'] [group_member_item = product_definition group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment [group_assignment.role -> group_role group_role.name = 'assembly'] [group_assignment.assigned_group -> group group.name = 'assembly definition']]}

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
assembly_ component_ relationship to geometric_model_ relationship_ with_ transformation (as transformation)	PATH			assembly_component_usage <= product_definition_usage <= product_definition_relationship characterized_product_definition = product_definition_relationship characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- context_dependent_shape_representation.represented_product_relation context_dependent_shape_representation context_dependent_shape_representation.representation_relation -> shape_representation_relationship <= representation_relationship => {[representation_relationship.rep_2 -> representation <- {representation => shape_representation} property_definition_representation.used_representation property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition <- product_definition_relationship.relating_product_definition] [representation_relationship.rep_1 ->

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation <- {representation => shape_representation} property_definition_representation.used_representation property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition <- product_definition_relationship.related_product_definition] product_definition_relationship => product_definition_usage => assembly_component_usage [1)]} representation_relationship_with_transformation
assembly_ component_ relationship to template_ instance (as transformation)	PATH			assembly_component_usage <= product_definition_usage <= product_definition_relationship ([product_definition_relationship.relating_product_definition -> product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item =>] [product_definition_relationship.related_product_definition -> product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <- {representation => shape_representation} representation_map.mapped_representation representation_map <-

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				mapped_item.mapping_source])[2)] ([product_definition_relationship.relating_product_definition -> product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item =>] [product_definition_relationship.related_product_definition -> product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <- {representation => shape_representation} representation_map.mapped_representation representation_map <- mapped_item.mapping_source] [product_definition_relationship characterized_product_definition = product_definition_relationship characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item =>])[3)] mapped_item {mapped_item.mapping_target -> representation_item => geometric_representation_item =>

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				(placement) (cartesian_transformation_operator cartesian_transformation_operator.scl = 1.0)}
ASSEMBLY_ DEFINITION	product_definition	41	32	{[product_definition product_definition.frame_of_reference ->] product_definition_context <= application_context_element application_context_element.name = 'definitional template'] [product_definition <- product_definition_relationship.relating_product_definition product_definition_relationship => product_definition_usage => assembly_component_usage] [product_definition group_member_item = product_definition group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment {group_assignment.role ->] group_role group_role.name = 'assembly'} group_assignment.assigned_group ->] group group.name = 'assembly definition']}
assembly_type	product_definition. description	41		
ASSEMBLY_ SUBSTITUTE_ RELATIONSHIP	assembly_component_ usage_substitute	44		
description	assembly_component_ usage_substitute. description	44		

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
assembly_ substitute_ relationship to assembly_ component_ relationship (as base)	PATH			assembly_component_usage_substitute assembly_component_usage_substitute.base -> assembly_component_usage
assembly_ substitute_ relationship to multi_language_ string (as description)	PATH		89	assembly_component_usage_substitute multi_language_attribute_item = assembly_component_usage_substitute multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
assembly_ substitute_ relationship to assembly_ component_ relationship (as substitute)	PATH			assembly_component_usage_substitute assembly_component_usage_substitute.substitute -> assembly_component_usage
GENERAL_ITEM_ DEFINITION_ INSTANCE_ RELATIONSHIP	product_definition_ relationship	41		{[product_definition_relationship.related_product_definition -> product_definition => product_definition_occurrence] [product_definition_relationship.relating_product_definition -> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template']}
description	product_definition_ relationship. description	41		

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
name	product_definition_ relationship.id	41		
relation_type	product_definition_ relationship.name	41		
general_item_ definition_ instance_ relationship to multi_language_ string (as description)	PATH		144	product_definition_relationship multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
general_item_ definition_ instance_ relationship to multi_language_ string (as name)	PATH		144	product_definition_relationship multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'id'}
GENERAL_ITEM_ DEFINITION_ RELATIONSHIP	product_definition_ relationship	41		{product_definition_relationship [product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
description	product_definition_ relationship. description	41		
name	product_definition_ relationship.id	41		
relation_type	product_definition_ relationship.name	41		

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
general_item_ definition_ relationship to multi_language_ string (as description)	PATH		144	product_definition_relationship multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
general_item_ definition_ relationship to multi_language_ string (as name)	PATH		144	product_definition_relationship multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'id'}
GENERAL_ITEM_ INSTANCE_ RELATIONSHIP	product_definition_ relationship	41		{product_definition_relationship [product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}
description	product_definition_ relationship. description	41		
name	product_definition_ relationship.id	41		
relation_type	product_definition_ relationship.name	41		
general_item_ instance_ relationship to multi_language_ string (as description)	PATH		144	product_definition_relationship multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
general_item_ instance_ relationship to multi_language_ string (as name)	PATH		144	product_definition_relationship multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'id'}
GEOMETRICAL_ RELATIONSHIP	product_definition_ relationship	41		{product_definition_relationship product_definition_relationship.name = 'geometrical relationship' }
description	product_definition_ relationship. description	41		
name	product_definition_ relationship.id	41		
geometrical_ relationship to multi_language_ string (as description)	PATH		144	product_definition_relationship {product_definition_relationship.name = 'geometrical relationship' } multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
geometrical_ relationship to multi_language_ string (as name)	PATH		144	product_definition_relationship {product_definition_relationship.name = 'geometrical relationship' } multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'id'}

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
geometrical_ relationship to geometric_model_ relationship_ with_ transformation (as transformation)	PATH			product_definition_relationship {product_definition_relationship.name = 'geometrical relationship' } characterized_product_definition = product_definition_relationship characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- context_dependent_shape_representation.represented_product_relation context_dependent_shape_representation context_dependent_shape_representation.representation_relation -> shape_representation_relationship <= representation_relationship => {representation_relationship [representation_relationship.rep_1 -> representation <- {representation => shape_representation} property_definition_representation.used_representation property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition <- product_definition_relationship.relating_product_definition] [representation_relationship.rep_2 ->

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation <- {representation => shape_representation} property_definition_representation.used_representation property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition <- product_definition_relationship.related_product_definition] product_definition_relationship} representation_relationship_with_transformation
geometrical_ relationship to template_ instance (as transformation)	PATH			product_definition_relationship {product_definition_relationship.name = 'geometrical relationship' } [product_definition_relationship.related_product_definition -> product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <-

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item =>] [product_definition_relationship.relating_product_definition -> product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <- {representation => shape_representation} representation_map.mapped_representation representation_map <- mapped_item.mapping_source] mapped_item {mapped_item.mapping_target ->

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item => geometric_representation_item => (placement) (cartesian_transformation_operator cartesian_transformation_operator.scl = 1.0)}
ITEM_DEFINITION_ INSTANCE_ RELATIONSHIP	product_definition_ relationship	41		{[(product_definition_relationship product_definition_relationship.related_product_definition ->> product_definition => product_definition_occurrence) (product_definition_relationship => {(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')} product_definition_usage => assembly_component_usage)] [product_definition_relationship product_definition_relationship.relating_product_definition ->> product_definition product_definition.frame_of_reference ->> product_definition_context <= application_context_element application_context_element.name = 'definitional template']}
item_definition_ instance_ relationship to item_instance (as related)	PATH			(product_definition_relationship product_definition_relationship.related_product_definition ->> product_definition => {product_definition.frame_of_reference ->> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'} product_definition_occurrence) (product_definition_relationship => {(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage')

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				(product_definition_relationship.name = 'single instance usage')} product_definition_usage => assembly_component_usage)
item_definition_ instance_ relationship to design_ discipline_item_ definition (as relating)	PATH			product_definition_relationship product_definition_relationship.relating_product_definition ->> product_definition {product_definition.frame_of_reference ->> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
ITEM_DEFINITION_ RELATIONSHIP	product_definition_ relationship	41		{product_definition_relationship [product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference ->> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
item_definition_ relationship to design_ discipline_item_ definition (as related)	PATH			product_definition_relationship product_definition_relationship.related_product_definition ->> product_definition {product_definition.frame_of_reference ->> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
item_definition_ relationship to design_ discipline_item_ definition (as relating)	PATH			product_definition_relationship product_definition_relationship.relating_product_definition ->> product_definition {product_definition.frame_of_reference ->> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
ITEM_INSTANCE	(product_definition_ occurrence) (assembly_component_ usage)	214 44		(product_definition_occurrence <= product_definition {[product_definition.frame_of_reference ->> product_definition_context <=

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) ({assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})
description	(product_definition. description) (product_definition_ relationship. description)	41 41		(product_definition_occurrence <= product_definition product_definition.description) (assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.description)
id	(product_definition. id) (product_definition_ relationship.id)	41 41		(product_definition_occurrence <= product_definition product_definition.id) (assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.id)
item_instance to design_ discipline_item_ definition (as definition)	PATH			(product_definition_occurrence <= product_definition <- product_definition_relationship.related_product_definition product_definition_relationship {product_definition_relationship.name = 'definition usage'} product_definition_relationship.relating_product_definition ->) (assembly_component_usage <=

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_usage <= product_definition_relationship product_definition_relationship.related_product_definition ->) product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
item_instance to product_specification (as definition)	PATH		8	product_definition_occurrence <= product_definition configuration_design_item = product_definition configuration_design_item <- configuration_design.design configuration_design {configuration_design => occurrence_usage_definition} configuration_design.configuration -> configuration_item => {configuration_item.description = 'product specification'} configurable_item
item_instance to multi_language_string (as description)	PATH		144,88, 143	(product_definition_occurrence multi_language_attribute_item = product_definition_occurrence multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'product_definition.description'}) (assembly_component_usage multi_language_attribute_item = assembly_component_usage multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				{applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'product_definition_relationship.description'})
ITEM_INSTANCE_ RELATIONSHIP	(product_definition_ relationship) (product_definition_ substitute)	41 41		{product_definition_relationship [product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}
item_instance_ relationship to item_instance (as related)	PATH			(product_definition_relationship product_definition_relationship.related_product_definition -> product_definition_occurrence) (product_definition_substitute product_definition_substitute.substitute_definition -> (product_definition <- product_definition_relationship.related_product_definition product_definition_relationship => {product_definition_relationship.relating_product_definition -> product_definition <- product_definition_relationship.relating_product_definition product_definition_relationship <- {product_definition_relationship => product_definition_usage => assembly_component_usage} product_definition_substitute.context_relationship[4)]} product_definition_usage => assembly_component_usage) (product_definition => product_definition_occurrence))

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
item_instance_ relationship to item_instance (as relating)	PATH			(product_definition_relationship product_definition_relationship.relating_product_definition ->\ product_definition => product_definition_occurrence) (product_definition_substitute product_definition_substitute.context_relationship ->\ (product_definition_relationship => product_definition_usage => assembly_component_usage) (product_definition_relationship product_definition_relationship.related_product_definition ->\ product_definition => product_definition_occurrence))
NEXT_HIGHER_ ASSEMBLY	next_assembly_usage_ occurrence	44	9	
PROMISSORY_USAGE	promissory_usage_ occurrence	44		
QUANTIFIED_ INSTANCE	(product_definition_ occurrence) (quantified_assembly_ component_usage)	214 44		({product_definition_occurrence <= product_definition product_definition.name = 'quantified instance'}) ({quantified_assembly_component_usage <= assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'quantified instance usage'})
quantified_ instance to value_with_unit (as quantity)	PATH			(product_definition_occurrence <= product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition.description = 'occurrence quantity'} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->\ representation {representation.description = 'quantity'} representation.items[1] ->\ representation_item =>\ {representation_item.name = 'quantity measure'} measure_representation_item <=\ measure_with_unit) (quantified_assembly_component_usage quantified_assembly_component_usage.quantity ->\ measure_with_unit)
SELECTED_ INSTANCE	(product_definition_ occurrence) (assembly_component_ usage)	214 44		({product_definition_occurrence <=\ product_definition product_definition.description = 'selected instance'}) ({assembly_component_usage <=\ product_definition_usage <=\ product_definition_relationship product_definition_relationship.name = 'selected instance usage'})
selection_ control	descriptive_ representation_item	45		(product_definition_occurrence <=\ product_definition characterized_product_definition = product_definition) (assembly_component_usage <=\ product_definition_usage <=\ product_definition_relationship characterized_product_definition = product_definition_relationship) characterized_definition = characterized_product_definition characterized_definition <-

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {representation.description = 'selection control'} representation.items[i] -> representation_item => {representation_item.name = 'selection condition'} descriptive_representation_item
selected_ instance to value_with_unit (as maximum_ quantity)	PATH			(product_definition_occurrence <= product_definition {product_definition.description = 'selected instance'} characterized_product_definition = product_definition) (assembly_component_usage <= product_definition_usage <= product_definition_relationship {product_definition_relationship.name = 'selected instance usage'} characterized_product_definition = product_definition_relationship) characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {representation.description = 'selection control'} representation.items[i] ->

Table 25 – Mapping table item_definition_structure UoF (S3) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation_item => {representation_item.name = 'maximum quantity'} measure_representation_item <= measure_with_unit
selected_ instance to value_with_unit (as minimum_ quantity)	PATH			(product_definition_occurrence <= product_definition characterized_product_definition = product_definition) (assembly_component_usage <= product_definition_usage <= product_definition_relationship characterized_product_definition = product_definition_relationship) characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {representation.description = 'selection control'} representation.items[i] -> representation_item => {representation_item.name = 'minimum quantity'} measure_representation_item <= measure_with_unit
SINGLE_INSTANCE	(product_definition_ occurrence) (assembly_component_ usage)	214 44		{(product_definition_occurrence <= product_definition product_definition.description = 'single instance') (assembly_component_usage <= product_definition_usage <= product_definition_relationship

Table 25 – Mapping table item_definition_structure UoF (S3) (concluded)

Application element	AIM element	Source	Rules	Reference path
				product_definition_relationship.name = 'single instance usage')}
VALUE_WITH_UNIT	measure_with_unit	41	35	
unit_component	(named_unit) (derived_unit)	41 41	25	measure_with_unit measure_with_unit.unit_component ->(unit = named_unit named_unit) (unit = derived_unit derived_unit)
value_component	measure_value	41		measure_with_unit measure_with_unit.value_component ->measure_value

NOTE – The following note numbers correspond to the footnotemarks listed in the table above:

1 Same instance as beginning of reference path.

2 This case has to be taken in case the component (related_product_definition) is used only once within the assembly (relating_product_definition) so that the association of a particular mapped_-item with the particular product_definition_relationship of which it describes the transformation is possible.

3 This case has to be taken in case the component (related_product_definition) is used multiple times within the assembly (relating_product_definition) in order to allow the association of a particular mapped_item with the particular product_definition_relationship of which it describes the transformation.

4 Same instance as beginning of OR.

Table 26 – Mapping table effectivity UoF (S4)

Application element	AIM element	Source	Rules	Reference path
EFFECTIVITY	<dated_effectivity> <event_based_ effectivity>	41 214	58,193	event_based_effectivity <= effectivity
description	effectivity. description	41		< dated_effectivity <= > < event_based_effectivity <= > effectivity effectivity.description
effectivity_ context	product_definition_ context.life_cycle_ stage	41	28	< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => {effectivity_assignment.role -> effectivity_role effectivity_role.name = 'effectivity context'} effective_context_assignment => applied_effective_context_assignment applied_effective_context_assignment.items[1] -> effective_context_item effective_context_item = product_definition_context product_definition_context product_definition_context.life_cycle_stage
role	effectivity_role. name	41	28	< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment {effectivity_assignment => effective_item_assignment => applied_effective_item_assignment} effectivity_assignment.role -> effectivity_role effectivity_role.name {(effectivity_role.name) (effectivity_role.name = 'actual') (effectivity_role.name = 'planned')

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				(effectivity_role.name = 'required')}
effectivity to organization (as concerned_ organization)	PATH		28	< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => {effectivity_assignment.role -> effectivity_role effectivity_role.name = 'concerned organization'} effective_context_assignment => applied_effective_context_assignment applied_effective_context_assignment.items[i] -> effective_context_item effective_context_item = organization organization
effectivity to multi_language_ string (as description)	PATH		109	< dated_effectivity <= > < event_based_effectivity <= > effectivity multi_language_attribute_item = effectivity multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
effectivity to date_time (as end_ definition)	(dated_effectivity. effectivity_end_ date) (PATH)	41		event_based_effectivity (date_item = event_based_effectivity date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment {date_assignment.role -> date_role date_role.name = 'effectivity end'} date_assignment.assigned_date ->

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				date => calendar_date) (date_and_time_item = event_based_effectivity date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment {date_and_time_assignment.role -> date_time_role date_time_role.name = 'effectivity end'} date_and_time_assignment.assigned_date_and_time -> date_and_time)
effectivity to event_reference (as end_ definition)	PATH		13	event_based_effectivity event_item = event_based_effectivity event_item <- event_assignment.items[i] event_assignment <= action_assignment {action_assignment.role -> action_role action_role.name = 'effectivity end'} action_assignment.assigned_action -> action => event_reference
effectivity to value_with_unit (as end_ definition)	PATH		13	event_based_effectivity event_item = event_based_effectivity event_item <- [event_assignment.items[i] event_assignment <= action_assignment {action_assignment.role -> action_role action_role.name = 'effectivity start'} action_assignment.assigned_action -> action <-

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				{action => event_reference} action_relationship.relating_action] [event_assignment.items[i] event_assignment <= action_assignment {action_assignment.role -> action_role action_role.name = 'effectivity end'} action_assignment.assigned_action -> action <- {action => event_reference} action_relationship.related_action] action_relationship {action_relationship.name = 'event relationship'} characterized_action_definition = action_relationship characterized_action_definition <- action_property.definition action_property <- {action_property.name = 'duration'} action_property_representation.property action_property_representation action_property_representation.representation -> representation representation.items[1] -> representation_item => {representation_item.name = 'duration'} measure_representation_item <= measure_with_unit {measure_with_unit => time_measure_with_unit}

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to assembly_ substitute_ relationship (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = assembly_component_usage_substitute assembly_component_usage_substitute
effectivity to class_category_ association (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = product_concept_feature_category_usage product_concept_feature_category_usage {product_concept_feature_category_usage <= group_assignment group_assignment.role -> group_role (group_role.name = 'mandatory category usage') (group_role.name = 'optional category usage')}
effectivity to class_condition_ association (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] ->

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				effective_item effective_item = product_concept_feature_association product_concept_feature_association
effectivity to class_function_ association (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = configuration_design configuration_design {configuration_design.configuration -> configuration_item configuration_item.description = 'class function definition'}
effectivity to class_inclusion_ association (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = product_concept_feature_association product_concept_feature_association {product_concept_feature_association.name = 'inclusion association'}
effectivity to class_ specification_ association (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_effective_item_assignment.items[i] -> effective_item effective_item = product_concept_feature_association product_concept_feature_association
effectivity to dated_ configuration (as is_applied_ to)	IDENTICAL MAPPING			
effectivity to design_ constraint (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = representation representation {representation.description = 'design constraint'}
effectivity to drawing (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = auto_design_drawing_revision auto_design_drawing_revision

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to external_ document (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item (effective_item = document document) (effective_item = physically_modelled_product_definition physically_modelled_product_definition)
effectivity to geometric_model (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item =shape_representation shape_representation
effectivity to item (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = product product

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to item_definition_ instance_ relationship (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = product_definition_relationship product_definition_relationship {[(product_definition_relationship.related_product_definition -> product_definition => product_definition_occurrence) (product_definition_relationship => {(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')} product_definition_usage => assembly_component_usage)] [product_definition_relationship.relating_product_definition -> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template']}
effectivity to item_definition_ relationship (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] ->

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				effective_item effective_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
effectivity to item_instance (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item (effective_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition [product_definition.frame_of_reference ->] product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (effective_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})
effectivity to item_instance_ relationship (as is_applied_ to)	PATH		68	< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] ->> effective_item (effective_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}) (effective_item = product_definition_substitute product_definition_substitute)
effectivity to item_solution (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] ->> effective_item effective_item = solution_based_product_definition solution_based_product_definition

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to item_version (as is_applied_to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = product_definition_formation product_definition_formation
effectivity to material (as is_applied_to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = material_designation material_designation
effectivity to process_plan (as is_applied_to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = process_plan process_plan

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to product_class (as is_applied_to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = product_class product_class
effectivity to product_component (as is_applied_to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = product_component product_component
effectivity to product_function (as is_applied_to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				application_context_element.name = 'functional definition'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}
effectivity to product_ specification (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = configurable_item configurable_item {configurable_item <= configuration_item configuration_item.description = 'product specification'}
effectivity to property (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item (effective_item = property_definition property_definition) (effective_item = action_property action_property)

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to security_ classification (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = security_classification security_classification
effectivity to simulation_ information (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = kinematic_property_definition kinematic_property_definition
effectivity to specification (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to specification_ category (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category)
effectivity to specification_ expression (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item effective_item = conditional_concept_feature conditional_concept_feature {conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'expression'}
effectivity to specification_ inclusion (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] -> effective_item

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				effective_item = conditional_concept_feature conditional_concept_feature {[conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'inclusion'] [conditional_concept_feature.condition ->] concept_feature_relationship_with_condition concept_feature_relationship_with_condition.conditional_operator = 'implication']}
effectivity to styled_model (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] ->] effective_item (effective_item = draughting_model draughting_model) (effective_item = mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)
effectivity to view_area (as is_applied_ to)	PATH			< dated_effectivity <= > < event_based_effectivity <= > effectivity <- effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment applied_effective_item_assignment.items[i] ->] effective_item effective_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity to date_time (as start_ definition)	dated_effectivity. effectivity_start_ date	41		
effectivity to event_reference (as start_ definition)	PATH		13	event_based_effectivity event_item = event_based_effectivity event_item <- event_assignment.items[i] event_assignment <= action_assignment {action_assignment.role -> action_role action_role.name = 'effectivity start'} action_assignment.assigned_action -> action => event_reference
EFFECTIVITY_ RELATIONSHIP	effectivity_ relationship	41		
description	effectivity_ relationship. description	41		
relation_type	effectivity_ relationship.name	41		{effectivity_relationship (effectivity_relationship.name) (effectivity_relationship.name = 'constraint') (effectivity_relationship.name = 'inheritance')}
effectivity_ relationship to multi_language_ string (as description)	PATH		110	effectivity_relationship multi_language_attribute_item = effectivity_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
effectivity_ relationship to effectivity (as related)	PATH			effectivity_relationship effectivity_relationship.related_effectivity ->effectivity
effectivity_ relationship to effectivity (as relating)	PATH			effectivity_relationship effectivity_relationship.relating_effectivity ->effectivity
EVENT_REFERENCE	event_reference	214		event_reference <= action
event_type	action.name	41		event_reference <= action action.description
event_reference to activity (as event_ context)	PATH		13	event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role ->action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] ->event_context_item event_context_item = action action
event_reference to activity_ element (as event_ context)	PATH		13	event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role ->action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] ->event_context_item event_context_item = applied_action_assignment

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'}
event_reference to assembly_ substitute_ relationship (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = assembly_component_usage_substitute assembly_component_usage_substitute
event_reference to class_ category_ association (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product_concept_feature_category_usage product_concept_feature_category_usage {product_concept_feature_category_usage <= group_assignment group_assignment.role -> group_role (group_role.name = 'mandatory category usage') (group_role.name = 'optional category usage')}

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
event_reference to class_ condition_ association (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product_concept_feature_association product_concept_feature_association
event_reference to class_ function_ association (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = configuration_design configuration_design {configuration_design.configuration -> configuration_item configuration_item.description = 'class function definition'}
event_reference to class_ inclusion_ association (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] ->

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				event_context_item event_context_item = product_concept_feature_association product_concept_feature_association {product_concept_feature_association.name = 'inclusion association'}
event_reference to class_specification_association (as event_context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product_concept_feature_association product_concept_feature_association
event_reference to configuration (as event_context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item [event_context_item = configured_effective_item_assignment configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}] [event_context_item = configured_effective_context_assignment configured_effective_context_assignment

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				{configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}]
event_reference to design_ constraint (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = representation representation {representation.description = 'design constraint'}
event_reference to design_ discipline_item_ definition (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product_definition product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
event_reference to drawing (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = auto_design_drawing_revision auto_design_drawing_revision
event_reference to external_ document (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item (event_context_item = document document) (event_context_item = physically_modelled_product_definition physically_modelled_product_definition)
event_reference to geometric_ model (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				event_context_item = shape_representation shape_representation
event_reference to item (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product product
event_reference to item_ definition_ instance_ relationship (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product_definition_relationship product_definition_relationship {[(product_definition_relationship.related_product_definition -> product_definition => product_definition_occurrence) (product_definition_relationship => {(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')} product_definition_usage => assembly_component_usage)]

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				[product_definition_relationship.relating_product_definition -> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template']}
event_reference to item_ definition_ relationship (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
event_reference to item_instance (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				(event_context_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition [product_definition.frame_of_reference ->] product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (event_context_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})
event_reference to item_instance_ relationship (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role ->] action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] ->] event_context_item (event_context_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence})

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				(event_context_item = product_definition_substitute product_definition_substitute)
event_reference to item_solution (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = solution_based_product_definition solution_based_product_definition
event_reference to item_version (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product_definition_formation product_definition_formation
event_reference to material (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] ->

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				event_context_item event_context_item = material_designation material_designation
event_reference to physical_ instance (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}
event_reference to physical_ instance_usage (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'physical occurrence usage'}

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
event_reference to process_plan (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = process_plan process_plan
event_reference to product_class (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product_class product_class
event_reference to product_ component (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product_component product_component

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
event_reference to product_ function (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}
event_reference to product_ specification (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = configurable_item configurable_item

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				{configurable_item <= configuration_item configuration_item.description = 'product specification'}
event_reference to project (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = project project
event_reference to property (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item (event_context_item = property_definition property_definition) (event_context_item = action_property action_property)
event_reference to simulation_ information (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'}

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = kinematic_property_definition kinematic_property_definition
event_reference to specification (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)
event_reference to specification_ category (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category)

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
event_reference to specification_ expression (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = conditional_concept_feature conditional_concept_feature {conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'expression'}
event_reference to specification_ inclusion (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = conditional_concept_feature conditional_concept_feature {[conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'inclusion'] [conditional_concept_feature.condition -> concept_feature_relationship_with_condition concept_feature_relationship_with_condition.conditional_operator = 'implication']}

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
event_reference to styled_model (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item (event_context_item = draughting_model draughting_model) (event_context_item = mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)
event_reference to view_area (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment event_context_assignment.items[i] -> event_context_item event_context_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)
event_reference to work_request (as event_ context)	PATH			event_reference <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'event context'} event_context_assignment

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				event_context_assignment.items[i] -> event_context_item event_context_item = versioned_action_request versioned_action_request
event_reference to value_with_ unit (as offset)	PATH			event_reference <= action characterized_action_definition = action characterized_action_definition <- action_property.definition action_property <- action_property_representation.property action_property_representation action_property_representation.representation -> representation {representation.description = 'offset definition'} representation.items[1] -> representation_item => {representation_item.name = 'offset value'} measure_representation_item <= measure_with_unit {measure_with_unit => time_measure_with_unit}
RETENTION_PERIOD	retention	214	183	retention <= action {action <- action_assignment.assigned_action action_assignment => retention_assignment}
retention_ purpose	action.description	41		retention <= action action.description
retention_period to date_time (as earliest_end_ definition)	PATH			retention (date_item = retention date_item <- applied_date_assignment.items[i]

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_date_assignment <= date_assignment date_assignment.assigned_date ->>>date =>>>{date <- date_assignment.assigned_date date_assignment date_assignment.role ->>>date_role date_role.name = 'earliest end'} calendar_date) (date_and_time_item = retention date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.assigned_date_and_time ->>>date_and_time {date_and_time <- date_and_time_assignment.assigned_date_and_time date_and_time_assignment date_and_time_assignment.role ->>>date_time_role date_time_role.name = 'earliest end'})
retention_period to event_reference (as earliest_end_definition)	PATH			retention <= action <- action_relationship.relating_action action_relationship {action_relationship.name = 'earliest end'} action_relationship.related_action ->>>action =>>>event_reference

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
retention_period to value_with_ unit (as earliest_end_ definition)	PATH			retention <= action characterized_action_definition = action characterized_action_definition <- action_property.definition action_property <- {action_property.name = 'retention period'} action_property_representation.property action_property_representation action_property_representation.representation -> representation representation.items[i] -> representation_item => {representation_item.name = 'earliest end'} measure_representation_item <= measure_with_unit {measure_with_unit => time_measure_with_unit}
retention_period to activity (as is_applied_ to)	PATH		13	retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = action action
retention_period to activity_ element (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = applied_action_assignment applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'}
retention_period to assembly_ substitute_ relationship (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = assembly_component_usage_substitute assembly_component_usage_substitute
retention_period to class_ category_ association (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = product_concept_feature_category_usage product_concept_feature_category_usage {product_concept_feature_category_usage <=

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				group_assignment group_assignment.role ->& group_role (group_role.name = 'mandatory category usage') (group_role.name = 'optional category usage')}
retention_period to class_condition_association (as is_applied_to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role ->& action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] ->& retention_item retention_item = product_concept_feature_association product_concept_feature_association
retention_period to class_function_association (as is_applied_to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role ->& action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] ->& retention_item retention_item = configuration_design configuration_design {configuration_design.configuration ->& configuration_item configuration_item.description = 'class function definition'}

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
retention_period to class_ inclusion_ association (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = product_concept_feature_association product_concept_feature_association {product_concept_feature_association.name = 'inclusion association'}
retention_period to class_ specification_ association (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = product_concept_feature_association product_concept_feature_association
retention_period to configuration (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				[retention_item = configured_effective_item_assignment configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}] [retention_item = configured_effective_context_assignment configured_effective_context_assignment {configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}]
retention_period to design_ constraint (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = representation representation {representation.description = 'design constraint'}
retention_period to design_ discipline_item_ definition (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				retention_assignment.items[i] -> retention_item retention_item = product_definition product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
retention_period to drawing (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = auto_design_drawing_revision auto_design_drawing_revision
retention_period to external_ document (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item (retention_item = document document) (retention_item = physically_modelled_product_definition physically_modelled_product_definition)

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
retention_period to geometric_model (as is_applied_to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = shape_representation shape_representation
retention_period to item (as is_applied_to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = product product
retention_period to item_definition_instance_relationship (as is_applied_to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = product_definition_relationship product_definition_relationship

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				{[(product_definition_relationship.related_product_definition -> product_definition => product_definition_occurrence) (product_definition_relationship => {(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')} product_definition_usage => assembly_component_usage)] [product_definition_relationship.relating_product_definition -> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template']}
retention_period to item_ definition_ relationship (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
retention_period to item_instance (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item (retention_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (retention_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})
retention_period to item_instance_ relationship (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item (retention_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}) (retention_item = product_definition_substitute product_definition_substitute)
retention_period to item_solution (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = solution_based_product_definition solution_based_product_definition
retention_period to item_version (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = product_definition_formation

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_formation
retention_period to material (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = material_designation material_designation
retention_period to physical_ instance (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}
retention_period to physical_ instance_usage (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'physical occurrence usage'}
retention_period to process_plan (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = process_plan process_plan
retention_period to product_class (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = product_class product_class

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
retention_period to product_ component (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = product_component product_component
retention_period to product_ function (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
retention_period to product_ specification (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = configurable_item configurable_item {configurable_item <= configuration_item configuration_item.description = 'product specification'}
retention_period to project (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = project project
retention_period to property (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] ->

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				retention_item (retention_item = property_definition property_definition) (retention_item = action_property action_property)
retention_period to simulation_ information (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = kinematic_property_definition kinematic_property_definition
retention_period to specification (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
retention_period to specification_ category (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category)
retention_period to specification_ expression (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = conditional_concept_feature conditional_concept_feature {conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'expression'}
retention_period to specification_ inclusion (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'}

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				retention_assignment retention_assignment.items[i] -> retention_item retention_item = conditional_concept_feature conditional_concept_feature {[conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'inclusion'] [conditional_concept_feature.condition -> concept_feature_relationship_with_condition concept_feature_relationship_with_condition.conditional_operator = \ 'implication']}
retention_period to styled_model (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item (retention_item = draughting_model draughting_model) (retention_item = mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)
retention_period to view_area (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				retention_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)
retention_period to work_request (as is_applied_ to)	PATH			retention <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role -> action_role action_role.name = 'retention'} retention_assignment retention_assignment.items[i] -> retention_item retention_item = versioned_action_request versioned_action_request
retention_period to date_time (as latest_end_ definition)	PATH			retention (date_item = retention date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment date_assignment.assigned_date -> date => {date <- date_assignment.assigned_date date_assignment date_assignment.role -> date_role date_role.name = 'latest end'} calendar_date) (date_and_time_item = retention date_and_time_item <- applied_date_and_time_assignment.items[i]

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.assigned_date_and_time ->>>date_and_time {date_and_time <- date_and_time_assignment.assigned_date_and_time date_and_time_assignment date_and_time_assignment.role ->>>date_time_role date_time_role.name = 'latest end'})
retention_period to event_reference (as latest_end_definition)	PATH			retention <= action <- action_relationship.relating_action action_relationship {action_relationship.name = 'latest end'} action_relationship.related_action ->>>action => event_reference
retention_period to value_with_unit (as latest_end_definition)	PATH			retention <= action characterized_action_definition = action characterized_action_definition <- action_property.definition action_property <- {action_property.name = 'retention period'} action_property_representation.property action_property_representation action_property_representation.representation ->>>representation representation.items[i] ->>>representation_item => {representation_item.name = 'latest end'} measure_representation_item <= measure_with_unit {measure_with_unit => time_measure_with_unit}

Table 26 – Mapping table effectivity UoF (S4) (continued)

Application element	AIM element	Source	Rules	Reference path
retention_period to multi_language_string (as retention_purpose)	PATH		157	retention multi_language_attribute_item = retention multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'action.description'}
retention_period to date_time (as start_definition)	PATH			retention (date_item = retention date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment date_assignment.assigned_date -> date => {date <- date_assignment.assigned_date date_assignment date_assignment.role -> date_role date_role.name = 'start'} calendar_date) (date_and_time_item = retention date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.assigned_date_and_time -> date_and_time {date_and_time <- date_and_time_assignment.assigned_date_and_time date_and_time_assignment date_and_time_assignment.role ->

Table 26 – Mapping table effectivity UoF (S4) (concluded)

Application element	AIM element	Source	Rules	Reference path
				date_time_role date_time_role.name = 'start'})
retention_period to event_ reference (as start_ definition)	PATH			retention <= action <- action_relationship.relating_action action_relationship {action_relationship.name = 'start'} action_relationship.related_action -> action => event_reference
VALUE_WITH_UNIT	measure_with_unit	41	35	{measure_with_unit => time_measure_with_unit}
unit_component	(named_unit) (derived_unit)	41 41	25	measure_with_unit measure_with_unit.unit_component -> (unit = named_unit named_unit {named_unit => time_unit}) (unit = derived_unit derived_unit)
value_component	measure_value	41		measure_with_unit measure_with_unit.value_component -> measure_value {measure_value = time_measure time_measure}

Table 27 – Mapping table work_management UoF (S5)

Application element	AIM element	Source	Rules	Reference path
ACTIVITY	action	41		
activity_type	action.name	41		{(action.name) (action.name = 'amendment') (action.name = 'analysis') (action.name = 'cancellation') (action.name = 'delivery change') (action.name = 'design') (action.name = 'design change') (action.name = 'mock-up creation') (action.name = 'prototype building') (action.name = 'rectification') (action.name = 'restructuring') (action.name = 'spare part creation') (action.name = 'stop notice') (action.name = 'testing')}
description	action.description	41		
id	action.id	41		
status	action_status.status	41		action => executed_action <- action_status.assigned_action action_status action_status.status
activity to date_time (as actual_end_date)	PATH			action (date_item = action date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment date_assignment.assigned_date -> date => {date <- date_assignment.assigned_date date_assignment date_assignment.role -> date_role

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				date_role.name = 'actual end'} calendar_date) (date_and_time_item = action date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.assigned_date_and_time -> date_and_time {date_and_time <- date_and_time_assignment.assigned_date_and_time date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'actual end'})
activity to date_time (as actual_start_date)	PATH			action (date_item = action date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment date_assignment.assigned_date -> date => {date <- date_assignment.assigned_date date_assignment date_assignment.role -> date_role date_role.name = 'actual start'} calendar_date) (date_and_time_item = action date_and_time_item <- applied_date_and_time_assignment.items[i]

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.assigned_date_and_time -> date_and_time {date_and_time <- date_and_time_assignment.assigned_date_and_time date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'actual start'})
activity to activity_method (as chosen_ method)	PATH			action action.chosen_method -> action_method
activity to organization (as concerned_ organization)	PATH			action organization_item = action organization_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment.role -> organization_role organization_role.name = 'concerned organization'} organization_assignment.assigned_organization -> organization
activity to multi_language_ string (as description)	PATH		79	action multi_language_attribute_item = action multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
activity to date_time (as planned_end_date)	PATH			action (date_item = action date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment date_assignment.assigned_date -> date => {date <- date_assignment.assigned_date date_assignment date_assignment.role -> date_role date_role.name = 'planned end'} calendar_date) (date_and_time_item = action date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.assigned_date_and_time -> date_and_time {date_and_time <- date_and_time_assignment.assigned_date_and_time date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'planned end'})
activity to event_reference (as planned_end_date)	PATH			action <- action_relationship.relating_action action_relationship {action_relationship.name = 'planned end'} action_relationship.related_action -> action =>

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				event_reference
activity to value_with_unit (as planned_end_date)	PATH			action characterized_action_definition = action characterized_action_definition <- action_property.definition action_property <- {action_property.name = 'duration'} action_property_representation.property action_property_representation action_property_representation.representation -> representation representation.items[i] -> representation_item => {representation_item.name = 'planned duration'} measure_representation_item <= measure_with_unit {measure_with_unit => time_measure_with_unit}
activity to date_time (as planned_start_date)	PATH			action (date_item = action date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment date_assignment.assigned_date -> date => {date <- date_assignment.assigned_date date_assignment date_assignment.role -> date_role date_role.name = 'planned start'} calendar_date) (date_and_time_item = action

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.assigned_date_and_time -> date_and_time {date_and_time <- date_and_time_assignment.assigned_date_and_time date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'planned start'})
activity to event_reference (as planned_ start_date)	PATH			action <- action_relationship.relating_action action_relationship {action_relationship.name = 'planned start'} action_relationship.related_action -> action => event_reference
activity to date_ and_person_or_ organization (as requestor)	PATH		63,39, 37	action (date_and_person_item = action date_and_person_item <- applied_date_and_person_assignment.items[i] applied_date_and_person_assignment {applied_date_and_person_assignment <= person_assignment person_assignment.role -> person_role person_role.name = 'requestor'}) (date_and_person_and_organization_item = action date_and_person_and_organization_item <- applied_date_and_person_and_organization_assignment.items[i] applied_date_and_person_and_organization_assignment {applied_date_and_person_and_organization_assignment <=

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				person_and_organization_assignment person_and_organization_assignment.role ->
person_and_organization_role person_and_organization_role.name = 'requestor'}) (date_and_organization_item = action date_and_organization_item <- applied_date_and_organization_assignment.items[i] applied_date_and_organization_assignment {applied_date_and_organization_assignment <= organization_assignment organization_assignment.role ->
organization_role organization_role.name = 'requestor'})
activity to work_request (as resolved_request)	PATH			(action => executed_action => directed_action directed_action.directive ->
action_directive action_directive.requests[i] ->) (action action.chosen_method ->
action_method <- action_request_solution.method action_request_solution action_request_solution.request ->) versioned_action_request
activity to organization (as supplying_organization)	PATH			action organization_item = action organization_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment.role ->
organization_role

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				organization_role.name = 'supplying organization'} organization_assignment.assigned_organization ->
organization
ACTIVITY_ELEMENT	applied_action_assignment	214	13	applied_action_assignment <=
action_assignment
{action_assignment.role ->
action_role
action_role.description = 'activity element'}
role	action_role.name	41	13	applied_action_assignment <=
action_assignment
action_assignment.role ->
action_role
action_role.name
{(action_role.name)
(action_role.name = 'input')
(action_role.name = 'output')}
activity_element to activity (as activity)	PATH			applied_action_assignment <=
action_assignment
{action_assignment.role ->
action_role
action_role.description = 'activity element'}
action_assignment.assigned_action ->
action
activity_element to activity_method (as element)	PATH			applied_action_assignment
{applied_action_assignment <=
action_assignment
action_assignment.role ->
action_role
action_role.description = 'activity element'}
applied_action_assignment.items[i] ->
action_item
action_item = action_method
action_method

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
activity_element to assembly_ substitute_ relationship (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = assembly_component_usage_substitute assembly_component_usage_substitute
activity_element to class_ category_ association (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = product_concept_feature_category_usage product_concept_feature_category_usage {product_concept_feature_category_usage <= group_assignment group_assignment.role -> group_role (group_role.name = 'mandatory category usage') (group_role.name = 'optional category usage')}
activity_element to class_ condition_ association (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = product_concept_feature_association product_concept_feature_association

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
activity_element to class_ function_ association (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = configuration_design configuration_design {configuration_design.configuration -> configuration_item configuration_item.description = 'class function definition'}
activity_element to class_ inclusion_ association (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = product_concept_feature_association product_concept_feature_association {product_concept_feature_association.name = 'inclusion association'}
activity_element to class_ specification_ association (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = product_concept_feature_association product_concept_feature_association

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
activity_element to configuration (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item [action_item = configured_effective_item_assignment configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}] [action_item = configured_effective_context_assignment configured_effective_context_assignment {configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}]
activity_element to design_ constraint (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = representation representation {representation.description = 'design constraint'}

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
activity_element to drawing (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = auto_design_drawing_revision auto_design_drawing_revision
activity_element to external_ document (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item (action_item = document document) (action_item = physically_modelled_product_definition physically_modelled_product_definition)
activity_element to geometric_ model (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = shape_representation shape_representation
activity_element to item (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role ->

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = product product
activity_element to item_ definition_ instance_ relationship (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = product_definition_relationship product_definition_relationship {[(product_definition_relationship.related_product_definition -> product_definition => product_definition_occurrence) (product_definition_relationship => {(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')} product_definition_usage => assembly_component_usage)] [product_definition_relationship.relating_product_definition -> product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template']}

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
activity_element to item_ definition_ relationship (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
activity_element to item_instance (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item (action_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]})

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				(action_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})
activity_element to item_instance_ relationship (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item (action_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence}) (action_item = product_definition_substitute product_definition_substitute)
activity_element to item_solution (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = solution_based_product_definition solution_based_product_definition

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
activity_element to item_version (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = product_definition_formation product_definition_formation
activity_element to material (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = material_designation material_designation
activity_element to physical_ instance (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
activity_element to physical_ instance_usage (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role ->\ action_role action_role.description = 'activity element'} applied_action_assignment.items[i] ->\ action_item action_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'physical occurrence usage'}
activity_element to process_plan (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role ->\ action_role action_role.description = 'activity element'} applied_action_assignment.items[i] ->\ action_item action_item = process_plan process_plan
activity_element to product_class (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role ->\ action_role action_role.description = 'activity element'} applied_action_assignment.items[i] ->\ action_item action_item = product_class product_class

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
activity_element to product_ component (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = product_component product_component
activity_element to product_ function (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
activity_element to product_ specification (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = configurable_item configurable_item {configurable_item <= configuration_item configuration_item.description = 'product specification'}
activity_element to property (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item (action_item = property_definition property_definition) (action_item = action_property action_property)
activity_element to simulation_ information (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = kinematic_property_definition kinematic_property_definition

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
activity_element to specification (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)
activity_element to specification_ category (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category)
activity_element to specification_ expression (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = conditional_concept_feature conditional_concept_feature {conditional_concept_feature <= product_concept_feature

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_concept_feature.name = 'expression'}
activity_element to specification_ inclusion (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = conditional_concept_feature conditional_concept_feature {[conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'inclusion'] [conditional_concept_feature conditional_concept_feature.condition -> concept_feature_relationship_with_condition concept_feature_relationship_with_condition.conditional_operator = \ 'implication']}
activity_element to styled_model (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item (action_item = draughting_model draughting_model) (action_item = mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)
activity_element to view_area (as element)	PATH			applied_action_assignment {applied_action_assignment <= action_assignment action_assignment.role -> action_role

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_role.description = 'activity element'} applied_action_assignment.items[i] -> action_item action_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)
ACTIVITY_METHOD	action_method	41		
consequence	action_method. consequence	41		
description	action_method. description	41		
name	action_method.name	41		
activity_method to multi_ language_string (as description)	PATH		81	action_method multi_language_attribute_item = action_method multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
activity_method to multi_ language_string (as name)	PATH		81	action_method multi_language_attribute_item = action_method multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
ACTIVITY_ RELATIONSHIP	action_relationship	41		
description	action_relationship. description	41		
relation_type	action_relationship. name	41		{(action_relationship.name) (action_relationship.name = 'decomposition') (action_relationship.name = 'precedence')

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				(action_relationship.name = 'derivation') (action_relationship.name = 'sequence')}
activity_ relationship to multi_language_ string (as description)	PATH		83	action_relationship multi_language_attribute_item = action_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
activity_ relationship to activity (as related)	PATH			action_relationship action_relationship.related_action -> action
activity_ relationship to activity (as relating)	PATH			action_relationship action_relationship.relating_action -> action
ALTERNATE_ITEM_ RELATIONSHIP	alternate_product_ relationship	44		
fulfilled_ requirements	alternate_product_ relationship.basis	44		
alternate_item_ relationship to item (as alternate)	PATH			alternate_product_relationship alternate_product_relationship.alternate -> product
alternate_item_ relationship to item (as base)	PATH			alternate_product_relationship alternate_product_relationship.base -> product

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
alternate_item_ relationship to multi_language_ string (as fulfilled_ requirements)	PATH		85	alternate_product_relationship multi_language_attribute_item = alternate_product_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'basis'}
CHANGE	(representation_ relationship) (product_definition_ formation_ relationship) (action_ relationship) (configuration_item_ relationship) (product_definition_ relationship) (shape_aspect_ relationship)	43 41 41 44 41 41		({representation_relationship [representation_relationship.name = 'change'] [[representation_relationship.rep_1 ->] [representation_relationship.rep_2 ->] representation representation.description = 'design constraint']}) ({product_definition_formation_relationship product_definition_formation_relationship.name = 'change'}) ({action_relationship [action_relationship.name = 'change'] [([action_relationship.related_action ->] [action_relationship.relating_action ->] action action.description = 'process operation') ([action_relationship.relating_action ->] [action_relationship.related_action ->] action => product_definition_process => process_plan)]}) ({configuration_item_relationship [configuration_item_relationship.name = 'change'] [[configuration_item_relationship.relating_configuration_item ->] [configuration_item_relationship.related_configuration_item ->] configuration_item => configuration_item_with_context =>

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_component]}) ({product_definition_relationship [product_definition_relationship.name = 'change'] [([product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition') ([product_definition_relationship.relating_product_definition ->] product_definition <- product_definition_relationship.relating_product_definition] [product_definition_relationship.related_product_definition ->] product_definition <- product_definition_relationship.related_product_definition] product_definition_relationship product_definition_relationship.name = 'definition replacement') ([product_definition_relationship.relating_product_definition ->] product_definition <- {product_definition =>] product_definition_occurrence} product_definition_relationship.related_product_definition product_definition_relationship <- product_definition_substitute.context_relationship] [product_definition_relationship.related_product_definition ->] product_definition <- {product_definition =>] product_definition_occurrence} product_definition_substitute.substitute_definition] product_definition_substitute)]}) ({shape_aspect_relationship

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship.name = 'change'}) 1)
description	(representation_ relationship. description) (product_definition_ formation_ relationship. description) (action_relationship. description) (configuration_item_ relationship. description) (product_definition_ relationship. description) (shape_aspect_ relationship. description)	43 41 41 44 41 41		
change to design_ constraint_ relationship (as described_ change)	IDENTICAL MAPPING			{representation_relationship [representation_relationship.name = 'change'] [[representation_relationship.rep_1 ->] [representation_relationship.rep_2 ->] representation representation.description = 'design constraint']}
change to item_ version_ relationship (as described_ change)	(IDENTICAL MAPPING) (PATH)			({product_definition_formation_relationship product_definition_formation_relationship.name = 'change'}) (product_definition_formation_relationship {product_definition_formation_relationship.name = 'change'} [product_definition_formation_relationship.\ relating_product_definition_formation -> product_definition_formation <- product_definition_formation_relationship.\

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				relating_product_definition_formation] [product_definition_formation_relationship.\ related_product_definition_formation ->>>>

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
change to product_function_ relationship (as described_ change)	IDENTICAL MAPPING			{product_definition_relationship [product_definition_relationship.name = 'change'] [[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition']}
change to replaced_ definition_ relationship (as described_ change)	PATH			product_definition_relationship {product_definition_relationship.name = 'change'} [product_definition_relationship.relating_product_definition -> product_definition <- {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'} product_definition_relationship.relating_product_definition] [product_definition_relationship.related_product_definition -> product_definition <- {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'} product_definition_relationship.related_product_definition] product_definition_relationship {product_definition_relationship.name = 'definition replacement'}
change to replaced_usage_ relationship (as described_ change)	PATH			product_definition_relationship {product_definition_relationship.name = 'change'} [product_definition_relationship.relating_product_definition -> product_definition <- product_definition_relationship.related_product_definition product_definition_relationship <- product_definition_substitute.context_relationship] [product_definition_relationship.related_product_definition -> product_definition <-

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_substitute.substitute_definition] product_definition_substitute
change to shape_ aspect_ relationship (as described_ change)	IDENTICAL MAPPING			{shape_aspect_relationship shape_aspect_relationship.name = 'change'}
change to multi_ language_string (as description)	PATH		155, 142, 83,97	(representation_relationship multi_language_attribute_item = representation_relationship) (product_definition_formation_relationship multi_language_attribute_item = \ product_definition_formation_relationship) (action_relationship multi_language_attribute_item = action_relationship) (configuration_item_relationship multi_language_attribute_item = configuration_item_relationship) (product_definition_relationship multi_language_attribute_item = product_definition_relationship) (shape_aspect_relationship multi_language_attribute_item = shape_aspect_relationship) multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
CONTRACT	contract	41		
id	contract.name	41		
ordered_price	contract.purpose	41		
contract to activity (as contracted_ element)	PATH		19	contract <- contract_assignment.assigned_contract contract_assignment => applied_contract_assignment applied_contract_assignment.items[i] -> contract_item contract_item = action

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				action
contract to item_version (as contracted_element)	PATH		19	contract <- contract_assignment.assigned_contract contract_assignment => applied_contract_assignment applied_contract_assignment.items[i] ->contract_item contract_item = product_definition_formation product_definition_formation
ELEMENT_DELIVERY	element_delivery	214		element_delivery <= directed_action
element_delivery to activity_element (as deliverable_element)	PATH		13	element_delivery <= directed_action <= executed_action <= action <- action_assignment.assigned_action action_assignment => {action_assignment.role ->action_role action_role.description = 'deliverable element'} element_delivery_assignment element_delivery_assignment.items[1] ->element_delivery_item element_delivery_item = applied_action_assignment applied_action_assignment {[applied_action_assignment <=action_assignment action_assignment.role ->action_role action_role.description = 'activity element'] [applied_action_assignment.items[1] ->action_item (action_item = product_definition_formation product_definition_formation) (action_item = product_definition product_definition

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				{[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}) (action_item = configurable_item configurable_item {configurable_item <= configuration_item configuration_item.description = 'product specification'})]}
element_delivery to organization (as destination)	PATH			element_delivery organization_item = element_delivery organization_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment.role -> organization_role organization_role.name = 'destination'} organization_assignment.assigned_organization -> organization
element_delivery to value_with_ unit (as quantity)	PATH			element_delivery <= directed_action <= executed_action <= action characterized_action_definition = action characterized_action_definition <- action_property.definition action_property <- action_property_representation.property action_property_representation action_property_representation.representation -> representation {representation.description = 'delivery quantity'} representation.items[1] -> representation_item =>

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				{representation_item.name = 'quantity'} measure_representation_item <= measure_with_unit
ITEM_VERSION_ RELATIONSHIP	product_definition_ formation_ relationship	41		
description	product_definition_ formation_ relationship. description	41		
relation_type	product_definition_ formation_ relationship.name	41		{product_definition_formation_relationship (product_definition_formation_relationship.name) (product_definition_formation_relationship.name = 'hierarchy') (product_definition_formation_relationship.name = 'sequence') (product_definition_formation_relationship.name = 'derivation') (product_definition_formation_relationship.name = 'alias')}
item_version_ relationship to multi_language_ string (as description)	PATH		142	product_definition_formation_relationship multi_language_attribute_item = product_definition_formation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
item_version_ relationship to item_version (as related)	PATH			product_definition_formation_relationship product_definition_formation_relationship.\ related_product_definition_formation -> product_definition_formation
item_version_ relationship to item_version (as relating)	PATH			product_definition_formation_relationship product_definition_formation_relationship.\ relating_product_definition_formation -> product_definition_formation
ORGANIZATION_IN_ CONTRACT	applied_organization_ assignment	214		applied_organization_assignment <= {applied_organization_assignment applied_organization_assignment.items[i] -> organization_item

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				organization_item = contract} organization_assignment {organization_assignment.role -> organization_role organization_role.description = 'organization in contract'}
role_of_ organization	organization_role. name			applied_organization_assignment <= organization_assignment organization_assignment.role -> organization_role organization_role.name
organization_in_ contract to contract (as contract)	PATH			applied_organization_assignment applied_organization_assignment.items[i] -> organization_item organization_item = contract contract
organization_in_ contract to organization (as contracted_ organization)	PATH			applied_organization_assignment <= organization_assignment organization_assignment.assigned_organization -> organization
organization_in_ contract to multi_language_ string (as role_of_ organization)	PATH		124	organization_role multi_language_attribute_item = organization_role multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
organization_in_ contract to date_ and_person_or_ organization (as signature)	PATH		39,37	applied_organization_assignment (date_and_person_item = applied_organization_assignment date_and_person_item <- applied_date_and_person_assignment.items[i] applied_date_and_person_assignment {applied_date_and_person_assignment <= person_assignment

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				person_assignment.role -> person_role person_role.name = 'signing for contract'}) (date_and_person_and_organization_item = applied_organization_assignment date_and_person_and_organization_item <- applied_date_and_person_and_organization_assignment.items[i] applied_date_and_person_and_organization_assignment {applied_date_and_person_and_organization_assignment <= person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'signing for contract'}) (date_and_organization_item = versioned_action_request date_and_organization_item <- applied_date_and_organization_assignment.items[i] applied_date_and_organization_assignment {applied_date_and_organization_assignment <= organization_assignment organization_assignment.role -> organization_role organization_role.name = 'signing for contract'})
ORGANIZATION_ RELATIONSHIP	organization_ relationship	41		
relation_type	organization_ relationship.name	41		{(organization_relationship.name) (organization_relationship.name = 'hierarchy') (organization_relationship.name = 'correspondance') (organization_relationship.name = 'succession')}
organization_ relationship to organization (as related)	PATH			organization_relationship organization_relationship.related_organization -> organization

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
organization_ relationship to organization (as relating)	PATH			organization_relationship organization_relationship.relating_organization ->organization
PROJECT	project	214		project <= organization
description	organization. description	41		project <= organization organization.description
id	organization.id	41		project <= organization organization.id
name	organization.name	41		project <= organization organization.name
project to date_ time (as actual_end_ date)	PATH			project (date_item = project date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment date_assignment.assigned_date ->date => {date <- date_assignment.assigned_date date_assignment date_assignment.role ->date_role date_role.name = 'actual end'} calendar_date) (date_and_time_item = project date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <=

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				date_and_time_assignment date_and_time_assignment.assigned_date_and_time ->date_and_time {date_and_time <-date_and_time_assignment.assigned_date_and_time date_and_time_assignment date_and_time_assignment.role ->date_time_role date_time_role.name = 'actual end'})
project to date_time (as actual_start_date)	PATH			project (date_item = project date_item <-applied_date_assignment.items[i] applied_date_assignment <=date_assignment date_assignment.assigned_date ->date => {date <-date_assignment.assigned_date date_assignment date_assignment.role ->date_role date_role.name = 'actual start'} calendar_date) (date_and_time_item = project date_and_time_item <-applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <=date_and_time_assignment date_and_time_assignment.assigned_date_and_time ->date_and_time {date_and_time <-date_and_time_assignment.assigned_date_and_time date_and_time_assignment

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				date_and_time_assignment.role -> date_time_role date_time_role.name = 'actual start'})
project to product_class (as affected_ product_class)	PATH			project <= organization <- organization_assignment.assigned_organization organization_assignment => {organization_assignment.role -> organization_role organization_role.name = 'affecting project'} project_assignment project_assignment.items[i] -> project_item project_item = product_class product_class
project to multi_ language_string (as description)	PATH		147	project multi_language_attribute_item = project multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'organization.description'}
project to multi_ language_string (as name)	PATH		147	project multi_language_attribute_item = project multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'organization.name'}

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
project to date_time (as planned_end_date)	PATH			project (date_item = project date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment date_assignment.assigned_date -> date => {date <- date_assignment.assigned_date date_assignment date_assignment.role -> date_role date_role.name = 'planned end'} calendar_date) (date_and_time_item = project date_and_time_item <- applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.assigned_date_and_time -> date_and_time {date_and_time <- date_and_time_assignment.assigned_date_and_time date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'planned end'})
project to event_reference (as planned_end_date)	PATH		13	project event_item = project event_item <- event_assignment.items[i] event_assignment <= action_assignment

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				{action_assignment.role -> action_role action_role.name = 'planned end'} action_assignment.assigned_action -> action => event_reference
project to value_ with_unit (as planned_end_ date)	PATH		13	project event_item = project event_item <- [event_assignment.items[i] event_assignment <= action_assignment {action_assignment.role -> action_role (action_role.name = 'planned start') (action_role.name = 'actual start')} action_assignment.assigned_action -> action <- {action => event_reference} action_relationship.relating_action] [event_assignment.items[i] event_assignment <= action_assignment {action_assignment.role -> action_role action_role.name = 'planned end'} action_assignment.assigned_action -> action <- {action => event_reference} action_relationship.related_action] action_relationship

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				{action_relationship.name = 'event relationship'} characterized_action_definition = action_relationship characterized_action_definition <- action_property.definition action_property <- {action_property.name = 'duration'} action_property_representation.property action_property_representation action_property_representation.representation -> representation representation.items[1] -> representation_item => {representation_item.name = 'planned duration'} measure_representation_item <= measure_with_unit {measure_with_unit => time_measure_with_unit}
project to date_ time (as planned_ start_date)	PATH			project (date_item = project date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment date_assignment.assigned_date -> date => {date <- date_assignment.assigned_date date_assignment date_assignment.role -> date_role date_role.name = 'planned start'} calendar_date) (date_and_time_item = project date_and_time_item <-

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_date_and_time_assignment.items[i] applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.assigned_date_and_time ->>>>

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
PROJECT_ RELATIONSHIP	organization_ relationship	41		
description	organization_ relationship. description	41		
relation_type	organization_ relationship.name	41		{organization_relationship (organization_relationship.name) (organization_relationship.name = 'sequence') (organization_relationship.name = 'decomposition') (organization_relationship.name = 'dependency') (organization_relationship.name = 'succession')}
project_ relationship to multi_language_ string (as description)	PATH		123	organization_relationship multi_language_attribute_item = organization_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
project_ relationship to project (as related)	PATH			organization_relationship organization_relationship.related_organization -> organization => project
project_ relationship to project (as relating)	PATH			organization_relationship organization_relationship.relating_organization -> organization => project
SIMULTANEOUS_ ACTIVITY	action_relationship	41		{action_relationship.name = 'simultaneity'}
description	action_relationship. description	41		

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
simultaneous_ activity to multi_language_ string (as description)	PATH		83	action_relationship multi_language_attribute_item = action_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
simultaneous_ activity to activity (as element)	PATH			action_relationship {action_relationship* action_relationship.name = 'simultaneity'} (action_relationship.relating_action ->) (action_relationship.related_action ->) action
VALUE_WITH_UNIT	measure_with_unit	41	35	
unit_component	(named_unit) (derived_unit)	41 41	25	measure_with_unit measure_with_unit.unit_component -> (unit = named_unit named_unit) (unit = derived_unit derived_unit)
value_component	measure_value	41		measure_with_unit measure_with_unit.value_component -> measure_value
WORK_ORDER	action_directive	41		
description	action_directive. comment	41		
id	action_directive. name	41		
internal	action_directive. analysis	41		{action_directive action_directive.analysis (action_directive.analysis = 'internal') (action_directive.analysis = 'external')}
version_id	identification_ assignment.assigned_ id	41	168,33	action_directive version_item = action_directive version_item <-

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				version_assignment.items[1] version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
work_order_type	action_directive. description	41		{action_directive (action_directive.description) (action_directive.description = 'design release') (action_directive.description = 'manufacturing release') (action_directive.description = 'management resolution') (action_directive.description = 'design deviation permit') (action_directive.description = 'production deviation permit')}
work_order to multi_language_ string (as description)	PATH		80	action_directive multi_language_attribute_item = action_directive multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'comment'}
work_order to activity (as is_ controlling)	PATH			action_directive <- directed_action.directive directed_action <= executed_action <= action
WORK_REQUEST	versioned_action_ request	41		
description	versioned_action_ request.description	41		
id	versioned_action_ request.id	41		
request_type	versioned_action_ request.purpose	41		{(versioned_action_request.purpose) (versioned_action_request.purpose = 'technical improvement') (versioned_action_request.purpose = 'government regulation')

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				(versioned_action_request.purpose = 'durability improvement') (versioned_action_request.purpose = 'tool improvement') (versioned_action_request.purpose = 'production relief') (versioned_action_request.purpose = 'production requirement') (versioned_action_request.purpose = 'quality improvement') (versioned_action_request.purpose = 'security reason') (versioned_action_request.purpose = 'standardization') (versioned_action_request.purpose = 'cost reduction') (versioned_action_request.purpose = 'customer rejection') (versioned_action_request.purpose = 'supplier request') (versioned_action_request.purpose = 'customer request') (versioned_action_request.purpose = 'change of standard') (versioned_action_request.purpose = 'production alignment') (versioned_action_request.purpose = 'procurement alignment')}
status	action_request_ status.status	41		versioned_action_request <- action_request_status.assigned_request action_request_status action_request_status.status
version_id	versioned_action_ request.version	41		
work_request to multi_language_ string (as description)	PATH		164	versioned_action_request multi_language_attribute_item = versioned_action_request multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
work_request to date_and_person_ or_organization (as notified_ person)	PATH		201,39, 37	versioned_action_request (date_and_person_item = versioned_action_request date_and_person_item <- applied_date_and_person_assignment.items[i] applied_date_and_person_assignment {applied_date_and_person_assignment <= person_assignment

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				person_assignment.role -> person_role person_role.name = 'notified person or organization'}) (date_and_person_and_organization_item = versioned_action_request date_and_person_and_organization_item <- applied_date_and_person_and_organization_assignment.items[i] applied_date_and_person_and_organization_assignment {applied_date_and_person_and_organization_assignment <= person_and_organization_assignment person_and_organization_assignment.role -> person_and_organization_role person_and_organization_role.name = 'notified person or organization'}) (date_and_organization_item = versioned_action_request date_and_organization_item <- applied_date_and_organization_assignment.items[i] applied_date_and_organization_assignment {applied_date_and_organization_assignment <= organization_assignment organization_assignment.role -> organization_role organization_role.name = 'notified person or organization'})
work_request to activity_method (as recommended_ method)	PATH	41		versioned_action_request <- action_request_solution.request action_request_solution {action_request_solution.description = 'recommended method'} action_request_solution.method -> action_method
work_request to date_and_person_ or_organization (as requestor)	PATH		1,39,37	versioned_action_request (date_and_person_item = versioned_action_request date_and_person_item <- applied_date_and_person_assignment.items[i] applied_date_and_person_assignment {applied_date_and_person_assignment <=

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				person_assignment person_assignment.role ->\ person_role person_role.name = 'requestor'}) (date_and_person_and_organization_item = versioned_action_request date_and_person_and_organization_item <- applied_date_and_person_and_organization_assignment.items[i] applied_date_and_person_and_organization_assignment {applied_date_and_person_and_organization_assignment <= person_and_organization_assignment person_and_organization_assignment.role ->\ person_and_organization_role person_and_organization_role.name = 'requestor'}) (date_and_organization_item = versioned_action_request date_and_organization_item <- applied_date_and_organization_assignment.items[i] applied_date_and_organization_assignment {applied_date_and_organization_assignment <= organization_assignment organization_assignment.role ->\ organization_role organization_role.name = 'requestor'})
work_request to activity_method (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role ->\ action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] ->\ action_request_item action_request_item = action_method action_method

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
work_request to assembly_ substitute_ relationship (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = assembly_component_usage_substitute assembly_component_usage_substitute
work_request to class_category_ association (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = product_concept_feature_category_usage product_concept_feature_category_usage {product_concept_feature_category_usage <= group_assignment group_assignment.role -> group_role (group_role.name = 'mandatory category usage') (group_role.name = 'optional category usage')}
work_request to class_condition_ association (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] ->

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_request_item action_request_item = product_concept_feature_association product_concept_feature_association
work_request to class_function_ association (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = configuration_design configuration_design {configuration_design.configuration -> configuration_item configuration_item.description = 'class function definition'}
work_request to class_inclusion_ association (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = product_concept_feature_association product_concept_feature_association {product_concept_feature_association.name = 'inclusion association'}
work_request to class_ specification_ association (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'}

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = product_concept_feature_association product_concept_feature_association
work_request to configuration (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item [action_request_item = configured_effective_item_assignment configured_effective_item_assignment {configured_effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}] [action_request_item = configured_effective_context_assignment configured_effective_context_assignment {configured_effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}]
work_request to design_ constraint (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'}

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_action_request_assignment applied_action_request_assignment.items[i] ->& action_request_item action_request_item = representation representation {representation.description = 'design constraint'}
work_request to drawing (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role ->& action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] ->& action_request_item action_request_item = auto_design_drawing_revision auto_design_drawing_revision
work_request to external_ document (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role ->& action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] ->& action_request_item (action_request_item = document document) (action_request_item = physically_modelled_product_definition physically_modelled_product_definition)
work_request to geometric_model (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role ->& action_request_role action_request_role.name = 'scope'}

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = shape_representation shape_representation
work_request to item (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = product product
work_request to item_definition_ instance_ relationship (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = product_definition_relationship product_definition_relationship {[(product_definition_relationship.related_product_definition -> product_definition => product_definition_occurrence) (product_definition_relationship => {(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')}

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_definition_usage => assembly_component_usage)] [product_definition_relationship.relating_product_definition ->] product_definition product_definition.frame_of_reference ->] product_definition_context <= application_context_element application_context_element.name = 'definitional template']}
work_request to item_definition_ relationship (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role ->] action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] ->] action_request_item action_request_item = product_definition_relationship product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference ->] product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
work_request to item_instance (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role ->] action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] ->

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_request_item (action_request_item = product_definition_occurrence product_definition_occurrence {product_definition_occurrence <= product_definition [product_definition.frame_of_reference ->] product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}) (action_request_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <= product_definition_relationship (product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')})
work_request to item_instance_ relationship (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role ->] action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] ->] action_request_item (action_request_item = product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => product_definition_occurrence})

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				(action_request_item = product_definition_substitute product_definition_substitute)
work_request to item_solution (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = solution_based_product_definition solution_based_product_definition
work_request to item_version (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = product_definition_formation product_definition_formation
work_request to material (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = material_designation material_designation

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
work_request to physical_ instance (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}
work_request to physical_ instance_usage (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = assembly_component_usage assembly_component_usage {assembly_component_usage <= product_definition_usage <- product_definition_relationship product_definition_relationship.name = 'physical occurrence usage'}
work_request to process_plan (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role ->

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = process_plan process_plan
work_request to product_class (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = product_class product_class
work_request to product_component (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = product_component product_component
work_request to product_function (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_action_request_assignment.items[i] -> action_request_item action_request_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}
work_request to product_ specification (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = configurable_item configurable_item {configurable_item <= configuration_item configuration_item.description = 'product specification'}
work_request to property (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item (action_request_item = property_definition property_definition) (action_request_item = action_property action_property)
work_request to simulation_ information (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = kinematic_property_definition kinematic_property_definition
work_request to specification (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = product_concept_feature (product_concept_feature) (product_concept_feature => package_product_concept_feature)

Table 27 – Mapping table work_management UoF (S5) (continued)

Application element	AIM element	Source	Rules	Reference path
work_request to specification_ category (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = product_concept_feature_category (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category)
work_request to specification_ expression (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item action_request_item = conditional_concept_feature conditional_concept_feature {conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'expression'}
work_request to specification_ inclusion (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role -> action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] -> action_request_item

Table 27 – Mapping table work_management UoF (S5) (concluded)

Application element	AIM element	Source	Rules	Reference path
				action_request_item = conditional_concept_feature conditional_concept_feature {[conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'inclusion'] [conditional_concept_feature.condition ->] concept_feature_relationship_with_condition concept_feature_relationship_with_condition.conditional_operator = \ 'implication']}
work_request to styled_model (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role ->] action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] ->] action_request_item (action_request_item = draughting_model draughting_model) (action_request_item = mechanical_design_geometric_presentation_representation mechanical_design_geometric_presentation_representation)
work_request to view_area (as scope)	PATH		12	versioned_action_request <- action_request_assignment.assigned_action_request action_request_assignment => {action_request_assignment.role ->] action_request_role action_request_role.name = 'scope'} applied_action_request_assignment applied_action_request_assignment.items[i] ->] action_request_item action_request_item = presentation_area (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)

NOTE – The following note numbers correspond to the footnotemarks listed in the table above:

1 In all cases, the relating (or rep_1) has been changed to the related (or rep_2).

Table 28 – Mapping table classification UoF (S6)

Application element	AIM element	Source	Rules	Reference path
CLASSIFICATION_ ATTRIBUTE	(representation) (externally_defined_ classification_ attribute)	43 214		({representation representation.description = 'classification attribute'}) (externally_defined_classification_attribute <= [representation] [externally_defined_item])
description	descriptive_ representation_item	45		(externally_defined_classification_attribute <= representation) (representation) {representation.description = 'classification attribute'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item
id	representation.id	43		externally_defined_classification_attribute <= representation representation.id
name	representation.name	43		externally_defined_classification_attribute <= representation representation.name
classification_ attribute to external_ document (as attribute_ source)	PATH		27	representation {representation.description = 'classification attribute'} document_reference_item = representation document_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference {document_reference.role -> document_role document_role.name = 'classification attribute source'} document_reference.assigned_document -> document

Table 28 – Mapping table classification UoF (S6) (continued)

Application element	AIM element	Source	Rules	Reference path
classification_ attribute to external_library_ reference (as attribute_ source)	PATH			externally_defined_classification_attribute <= externally_defined_item externally_defined_item.source -> external_source => known_source
classification_ attribute to property_value (as attribute_ value)	PATH			(representation <- {representation.description = 'classification attribute'}) (externally_defined_classification_attribute <= representation <-) representation_relationship.rep_1 representation_relationship {representation_relationship.name = 'classification attribute value'} representation_relationship.rep_2 -> representation => property_value_representation
classification_ attribute to multi_language_ string (as description)	PATH		101	(externally_defined_classification_attribute <= representation) (representation {representation.description = 'classification attribute'}) representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 28 – Mapping table classification UoF (S6) (continued)

Application element	AIM element	Source	Rules	Reference path
classification_ attribute to multi_language_ string (as name)	PATH		155,111, 153	(representation {representation.description = 'classification attribute'} multi_language_attribute_item = representation) (externally_defined_classification_attribute multi_language_attribute_item = externally_defined_classification_attribute) multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'representation.name'}
CLASSIFICATION_ SYSTEM	(classification_ system) (externally_defined_ classification_ system)	214 214		(classification_system <= group) (externally_defined_classification_system <= [group] [externally_defined_item])
description	group.description	41		(classification_system <=) (externally_defined_classification_system <=) group group.description
id	externally_defined_ item.item_id	41		externally_defined_classification_system <= externally_defined_item externally_defined_item.item_id
name	group.name	41		(classification_system <=) (externally_defined_classification_system <=) group group.name
classification_ system to external_ document (as classification_ system_source)	PATH		27	classification_system document_reference_item = classification_system document_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference {document_reference.role -> document_role

Table 28 – Mapping table classification UoF (S6) (continued)

Application element	AIM element	Source	Rules	Reference path
				document_role.name = 'classification system source'} document_reference.assigned_document ->
document
classification_ system to external_library_ reference (as classification_ system_source)	PATH			externally_defined_classification_system <=
externally_defined_item
externally_defined_item.source ->
external_source =>
known_source
classification_ system to multi_ language_string (as description)	PATH		94,112	(classification_system
multi_language_attribute_item = classification_system
multi_language_attribute_item <-)
(externally_defined_classification_system
multi_language_attribute_item = externally_defined_classification_system
multi_language_attribute_item <-)

applied_multi_language_attribute_assignment.items[1]
applied_multi_language_attribute_assignment
{applied_multi_language_attribute_assignment <=
multi_language_attribute_assignment
multi_language_attribute_assignment.attribute_name = 'group.description'} |
| classification_ system to multi_ language_string (as name) | PATH | | 94,112 | (classification_system
multi_language_attribute_item = classification_system
multi_language_attribute_item <-)
(externally_defined_classification_system
multi_language_attribute_item = externally_defined_classification_system
multi_language_attribute_item <-)

applied_multi_language_attribute_assignment.items[1]
applied_multi_language_attribute_assignment
{applied_multi_language_attribute_assignment <=
multi_language_attribute_assignment
multi_language_attribute_assignment.attribute_name = 'group.name'} |

Table 28 – Mapping table classification UoF (S6) (continued)

Application element	AIM element	Source	Rules	Reference path
GENERAL_ CLASSIFICATION	(characterized_ product_category) (externally_defined_ product_category)	214 214		(characterized_product_category <= [product_category] [characterized_object]) (externally_defined_product_category <= [externally_defined_item] [characterized_product_category <= [product_category] [characterized_object]])
name	product_category. name	41		(characterized_product_category <=) (externally_defined_product_category <= characterized_product_category <=) product_category product_category.name
general_ classification to classification_ attribute (as attribute)	PATH			(characterized_product_category <=) (externally_defined_product_category <= characterized_product_category <=) characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition {property_definition.description = 'category attribute'} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> (representation {representation.description = 'classification attribute'}) (representation => externally_defined_classification_attribute)

Table 28 – Mapping table classification UoF (S6) (continued)

Application element	AIM element	Source	Rules	Reference path
general_ classification to external_ document (as classification_ source)	PATH		27	characterized_product_category document_reference_item = characterized_product_category document_reference_item <- applied_document_reference.items[i] applied_document_reference <= document_reference {document_reference.role -> document_role document_role.name = 'classification source'} document_reference.assigned_document -> document
general_ classification to external_ library_ reference (as classification_ source)	PATH			externally_defined_product_category <= externally_defined_item externally_defined_item.source -> external_source => known_source
general_ classification to multi_ language_string (as name)	PATH		93,113	(characterized_product_category multi_language_attribute_item = characterized_product_category) (externally_defined_product_category multi_language_attribute_item = externally_defined_product_category) multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'product_category.name'}

Table 28 – Mapping table classification UoF (S6) (continued)

Application element	AIM element	Source	Rules	Reference path
general_ classification to classification_ system (as used_ classification_ system)	PATH		32	(characterized_product_category) (externally_defined_product_category <= characterized_product_category) group_member_item = characterized_product_category group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment {group_assignment.role -> group_role group_role.name = 'classified member'} group_assignment.assigned_group -> group => (classification_system) (externally_defined_classification_system)
GENERAL_ CLASSIFICATION_ HIERARCHY	product_category_ relationship	41		{product_category_relationship.name = 'general classification hierarchy'}
general_ classification_ hierarchy to general_ classification (as sub_ classification)	PATH			product_category_relationship product_category_relationship.sub_category -> product_category => (characterized_product_category) (characterized_product_category => externally_defined_product_category)
general_ classification_ hierarchy to general_ classification (as super_ classification)	PATH			product_category_relationship product_category_relationship.category -> product_category => (characterized_product_category) (characterized_product_category => externally_defined_product_category)
ITEM_ CLASSIFICATION	product_category	41		

Table 28 – Mapping table classification UoF (S6) (concluded)

Application element	AIM element	Source	Rules	Reference path
description	product_category. description	41		
item_ classification to item (as associated_ item)	PATH			product_category => product_related_product_category product_related_product_category.products[i] -> product
SPECIFIC_ CLASSIFICATION	product_category	41		
classification_ name	product_category. name	41		{product_category.name product_category <- product_category_relationship.sub_category}
classification_ type	product_category. name	41		product_category <- product_category_relationship.sub_category product_category_relationship product_category_relationship.category -> product_category product_category.name {(product_category.name) (product_category.name = 'application control') (product_category.name = 'assembly item') (product_category.name = 'completely knocked down item') (product_category.name = 'in process item') (product_category.name = 'prototype building') (product_category.name = 'regulated item') (product_category.name = 'safety item') (product_category.name = 'service item')}

Table 29 – Mapping table specification_control UoF (S7)

Application element	AIM element	Source	Rules	Reference path
CLASS_CATEGORY_ ASSOCIATION	product_concept_ feature_category_ usage	214		product_concept_feature_category_usage <= group_assignment {group_assignment.role -> group_role (group_role.name = 'mandatory category usage') (group_role.name = 'optional category usage')}
mandatory	group_assignment. role		32	product_concept_feature_category_usage <= group_assignment group_assignment.role -> group_role group_role.name {(group_role.name = 'mandatory category usage') (group_role.name = 'optional category usage')}
class_category_ association to specification_ category (as category)	PATH			product_concept_feature_category_usage <= group_assignment group_assignment.assigned_group -> group => (product_concept_feature_category) (product_concept_feature_category => exclusive_product_concept_feature_category)
class_category_ association to product_class (as product_ class)	PATH			product_concept_feature_category_usage product_concept_feature_category_usage.items[1] -> category_usage_item category_usage_item = product_class product_class
CLASS_CONDITION_ ASSOCIATION	product_concept_ feature_association	44		
condition_type	product_concept_ feature_association. name	44		{product_concept_feature_association (product_concept_feature_association.name) (product_concept_feature_association.name = 'part usage') (product_concept_feature_association.name = 'identification') (product_concept_feature_association.name = 'validity')}
description	product_concept_ feature_association. description	44		

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
class_condition_ association to specification_ expression (as condition)	PATH			product_concept_feature_association product_concept_feature_association.feature ->
product_concept_feature =>
{product_concept_feature.name = 'expression'}
conditional_concept_feature
class_condition_ association to multi_language_ string (as description)	PATH		138	product_concept_feature_association multi_language_attribute_item = product_concept_feature_association
multi_language_attribute_item <-
applied_multi_language_attribute_assignment.items[1]
applied_multi_language_attribute_assignment
{applied_multi_language_attribute_assignment <=
multi_language_attribute_assignment
multi_language_attribute_assignment.attribute_name = 'description'}
class_condition_ association to product_class (as product_ class)	PATH			product_concept_feature_association product_concept_feature_association.concept ->
product_concept =>
product_class
CLASS_FUNCTION_ ASSOCIATION	configuration_design	44		{configuration_design configuration_design.configuration ->
configuration_item
configuration_item.description = 'class function definition'}
description	configuration_design. description	44		
relation_type	configuration_design. name	44		{configuration_design (configuration_design.name)
(configuration_design.name = 'possibility')
(configuration_design.name = 'main function')}
class_function_ association to multi_language_ string (as description)	PATH		96	configuration_design {configuration_design.configuration ->
configuration_item
configuration_item.description = 'class function definition'}
multi_language_attribute_item = configuration_design
multi_language_attribute_item <-
applied_multi_language_attribute_assignment.items[1]
applied_multi_language_attribute_assignment

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				{applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
class_function_ association to product_class (as product_ class)	PATH			configuration_design configuration_design.configuration -> configuration_item {configuration_item.description = 'class function definition'} configuration_item.item_concept -> product_concept => product_class
class_function_ association to product_function (as product_ function)	PATH			configuration_design {configuration_design.configuration -> configuration_item configuration_item.description = 'class function definition'} configuration_design.design -> configuration_design_item configuration_design_item = product_definition product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}
CLASS_INCLUSION_ ASSOCIATION	product_concept_ feature_association	44		{product_concept_feature_association product_concept_feature_association.name = 'inclusion association'}
description	product_concept_ feature_association. description	44		

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
class_inclusion_ association to multi_language_ string (as description)	PATH		138	product_concept_feature_association multi_language_attribute_item = product_concept_feature_association multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
class_inclusion_ association to specification_ inclusion (as inclusion)	PATH			product_concept_feature_association product_concept_feature_association.feature -> product_concept_feature => {product_concept_feature.name = 'inclusion'} conditional_concept_feature
class_inclusion_ association to product_class (as product_ class)	PATH			product_concept_feature_association product_concept_feature_association.concept -> product_concept => product_class
CLASS_ SPECIFICATION_ ASSOCIATION	product_concept_ feature_association	44		
association_type	product_concept_ feature_association. name	44		{(product_concept_feature_association.name) (product_concept_feature_association.name = 'replaceable standard') (product_concept_feature_association.name = 'non replaceable standard') (product_concept_feature_association.name = 'availability') (product_concept_feature_association.name = 'identification') (product_concept_feature_association.name = 'option')}
class_ specification_ association to product_class (as product_ class)	PATH			product_concept_feature_association product_concept_feature_association.concept -> product_concept => product_class

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
class_ specification_ association to specification (as specification)	PATH			product_concept_feature_association product_concept_feature_association.feature -> (|product_concept_feature|) (product_concept_feature => package_product_concept_feature)
COLLECTED_ITEM_ ASSOCIATION	product_definition_ relationship	41		{[product_definition_relationship product_definition_relationship.name = 'collection membership'] [product_definition_relationship.relating_product_definition ->] product_definition group_member_item = product_definition group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment [group_assignment.role ->] group_role group_role.name = 'collection'] [group_assignment.assigned_group ->] group group.name = 'collection definition']]}
collected_item_ association to collection_ definition (as relating)	PATH			product_definition_relationship {product_definition_relationship.name = 'collection membership'} product_definition_relationship.relating_product_definition ->] product_definition {product_definition group_member_item = product_definition group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment [group_assignment.role ->] group_role group_role.name = 'collection'] [group_assignment.assigned_group ->] group

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				group.name = 'collection definition']}
COLLECTION_ DEFINITION	product_definition	41	32	{[product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'] [product_definition product_definition_relationship.relating_product_definition product_definition_relationship product_definition_relationship.name = 'collection membership'] [product_definition group_member_item = product_definition group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment [group_assignment.role -> group_role group_role.name = 'collection'] [group_assignment.assigned_group -> group group.name = 'collection definition']]}
purpose	product_definition. description	41		
collection_ definition to multi_language_ string (as purpose)	PATH		140	product_definition {product_definition group_member_item = product_definition group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment [group_assignment.role -> group_role group_role.name = 'collection']

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				[group_assignment.assigned_group -> group group.name = 'collection definition']} multi_language_attribute_item = product_definition multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
CONFIGURATION	[configured_ effective_item_ assignment] [configured_ effective_context_ assignment]	214 214	28	[configured_effective_item_assignment <= effective_item_assignment {effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}] [configured_effective_context_assignment <= effective_context_assignment {effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'}]
configuration to class_condition_ association (as is_solution_ for)	PATH			configured_effective_context_assignment {configured_effective_context_assignment <= effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'} configured_effective_context_assignment.items[i] -> configured_effective_context_item configured_effective_context_item = product_concept_feature_association product_concept_feature_association

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
configuration to class_ specification_ association (as is_solution_ for)	PATH			configured_effective_context_assignment {configured_effective_context_assignment <= effective_context_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'} configured_effective_context_assignment.items[i] -> configured_effective_context_item configured_effective_context_item = product_concept_feature_association product_concept_feature_association
configuration to item_solution (as usage)	PATH			configured_effective_item_assignment {configured_effective_item_assignment <= effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'} configured_effective_item_assignment.items[i] -> configured_effective_item configured_effective_item = solution_based_product_definition solution_based_product_definition
configuration to process_ operation (as usage)	PATH			configured_effective_item_assignment {configured_effective_item_assignment <= effective_item_assignment <= effectivity_assignment effectivity_assignment.role -> effectivity_role effectivity_role.name = 'configuration'} configured_effective_item_assignment.items[i] -> configured_effective_item configured_effective_item = action action {action.description = 'process operation'}

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
DATED_ CONFIGURATION	<dated_effectivity> <event_based_ effectivity>	41 214		<{dated_effectivity <= effectivity <- [effectivity_assignment.assigned_effectivity effectivity_assignment => effective_context_assignment => applied_effective_context_assignment] [effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment]}> <event_based_effectivity <= effectivity {effectivity <- [effectivity_assignment.assigned_effectivity effectivity_assignment => effective_context_assignment => applied_effective_context_assignment] [effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment]}>
DESCRIPTIVE_ SPECIFICATION	descriptive_ representation_item	45		
description	descriptive_ representation_item. description	45		
descriptive_ specification to multi_language_ string (as description)	PATH		101	descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
DESIGN_ CONSTRAINT	representation	43		{representation representation.description = 'design constraint'}
constraint_id	representation.id	43		
description	descriptive_ representation_item	45		representation {representation.description = 'design constraint'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item
name	representation.name	43		
design_ constraint to multi_language_ string (as description)	PATH		101	representation {representation.description = 'design constraint'} representation.items[i] -> representation_item => {representation_item.name = 'description'} descriptive_representation_item multi_language_attribute_item = descriptive_representation_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
design_ constraint to product_class (as is_valid_ for)	PATH			representation <- {representation.description = 'design constraint'} property_definition_representation.used_representation property_definition_representation {property_definition_representation.description = 'design constraint validity'} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object =>

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_class
design_ constraint to multi_language_ string (as name)	PATH		153	representation {representation.description = 'design constraint'} multi_language_attribute_item = representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
DESIGN_ CONSTRAINT_ ASSOCIATION	property_definition_ representation	41		{property_definition_representation property_definition_representation.description = 'design constraint association'}
name	property_definition_ representation.name	41		
design_ constraint_ association to design_ constraint (as is_based_on)	PATH			property_definition_representation {property_definition_representation.description = 'design constraint association'} property_definition_representation.used_representation -> representation {representation.description = 'design constraint'}
design_ constraint_ association to item_solution (as is_ constraining)	PATH			property_definition_representation {property_definition_representation.description = 'design constraint association'} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition => solution_based_product_definition

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
design_ constraint_ association to product_ component (as is_ constraining)	PATH			property_definition_representation {property_definition_representation.description = 'design constraint association'} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => product_component
design_ constraint_ association to multi_language_ string (as name)	PATH		151	property_definition_representation {property_definition_representation.description = 'design constraint association'} multi_language_attribute_item = property_definition_representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
DESIGN_ CONSTRAINT_ RELATIONSHIP	representation_ relationship	43		{representation_relationship [representation_relationship.rep_1 -> representation representation.description = 'design constraint'] [representation_relationship.rep_2 -> representation representation.description = 'design constraint']}
description	representation_ relationship. description	43		
relation_type	representation_ relationship.name	43		{(representation_relationship.name) (representation_relationship.name = 'decomposition') (representation_relationship.name = 'precedence')}

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
design_ constraint_ relationship to multi_language_ string (as description)	PATH		155	representation_relationship {[representation_relationship.rep_1 -> representation representation.description = 'design constraint'] [representation_relationship.rep_2 -> representation representation.description = 'design constraint']} multi_language_attribute_item = representation_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
design_ constraint_ relationship to design_ constraint (as related)	PATH			representation_relationship representation_relationship.rep_2 -> representation {representation.description = 'design constraint'}
design_ constraint_ relationship to design_ constraint (as relating)	PATH			representation_relationship representation_relationship.rep_1 -> representation {representation.description = 'design constraint'}
DESIGN_ CONSTRAINT_ VERSION	representation	43	180,33	{representation [representation.description = 'design constraint'] [version_item = representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role -> identification_role identification_role.name = 'version id']}

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
version_id	identification_ assignment.assigned_ id	41	180,33	representation {representation.description = 'design constraint'} version_item = representation version_item <- version_assignment.items[1] version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
FINAL_ITEM_ SOLUTION	solution_based_ product_definition	214		solution_based_product_definition <= product_definition {(product_definition.name = 'final') (product_definition.name = 'technical final') (product_definition.name = 'supplier final') (product_definition.name = 'technical supplier final')}
final_item_ status	descriptive_ representation_item	45		solution_based_product_definition <= product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {representation.description = 'final item characteristics'} representation.items[i] -> representation_item => {representation_item.name = 'final item status'} descriptive_representation_item

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
final_item_ solution to descriptive_ specification (as final_ specification)	PATH			solution_based_product_definition <= product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {representation.name = 'final item characteristics'} representation.items[i] -> representation_item => {representation_item.name = 'final specification'} descriptive_representation_item
final_item_ solution to design_ discipline_item_ definition (as final_ specification)	PATH			solution_based_product_definition <= product_definition <- product_definition_relationship.relating_product_definition product_definition_relationship {product_definition_relationship.name = 'final specification'} product_definition_relationship.related_product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
FUNCTION_ COMPONENT_ ASSOCIATION	component_function_ definition	214	8	component_function_definition <= configuration_design

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
function_ component_ association to product_ component (as product_ component)	PATH			component_function_definition <= configuration_design configuration_design.configuration -> configuration_item => configuration_item_with_context => product_component
function_ component_ association to product_function (as product_ function)	PATH			component_function_definition <= configuration_design configuration_design.design -> configuration_design_item configuration_design_item = product_definition product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'}
INSTANCE_ PLACEMENT	(representation_ relationship_with_ transformation) (mapped_item)	43 43		{representation_relationship_with_transformation <= representation_relationship representation_relationship.name = 'instance placement'}
instance_ placement to single_instance (as placed_ instance)	PATH			(representation_relationship_with_transformation <= representation_relationship {representation_relationship.name = 'instance placement'} representation_relationship.rep_1 ->) (mapped_item mapped_item.mapping_source -> representation_map representation_map.mapped_representation ->) representation <- {representation => shape_representation} property_definition_representation.used_representation property_definition_representation

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition characterized_definition = characterized_product_definition characterized_product_definition = product_definition product_definition => {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [product_definition.name = 'single instance']} product_definition_occurrence
instance_ placement to product_ component (as product_ component)	PATH			(representation_relationship_with_transformation <= representation_relationship {representation_relationship.name = 'instance placement'} representation_relationship.rep_2 ->) (mapped_item <= {mapped_item.mapping_target -> representation_item => geometric_representation_item => (placement) (cartesian_transformation_operator cartesian_transformation_operator.scl = 1.0)} representation_item <- representation.items[i]) representation <- {representation =>

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation} property_definition_representation.used_representation property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.definition -> represented_definition represented_definition = property_definition property_definition {property_definition => product_definition_shape} property_definition.definition -> characterized_definition characterized_definition = characterized_object characterized_object => product_component
instance_ placement to geometric_model_ relationship_ with_ transformation (as transformation)	IDENTICAL MAPPING			
instance_ placement to template_ instance (as transformation)	IDENTICAL MAPPING			{mapped_item mapped_item.mapping_target -> representation_item => geometric_representation_item => (placement) (cartesian_transformation_operator cartesian_transformation_operator.scl = 1.0)}
ITEM_SOLUTION	solution_based_ product_definition	214		solution_based_product_definition <= product_definition
id	product_definition. id	41		solution_based_product_definition <= product_definition product_definition.id

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
version_id	identification_ assignment.assigned_ id	41	182,33	solution_based_product_definition version_item = solution_based_product_definition version_item <- version_assignment.items[1] version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
item_solution to item_solution (as base_ element)	PATH			solution_based_product_definition <= product_definition <- product_definition_relationship.related_product_definition product_definition_relationship {product_definition_relationship.name = 'solution alternative definition'} product_definition_relationship.relating_product_definition -> product_definition => solution_based_product_definition
item_solution to product_ component (as base_ element)	PATH		8	solution_based_product_definition <= product_definition configuration_design_item = product_definition configuration_design_item <- configuration_design.design configuration_design {configuration_design => solution_alternative_definition} configuration_design.configuration -> configuration_item => configuration_item_with_context => product_component
ITEM_SOLUTION_ RELATIONSHIP	product_definition_ relationship	41		{product_definition_relationship [product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => solution_based_product_definition}

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
description	product_definition_ relationship. description	41		
relation_type	product_definition_ relationship.name	41		{(product_definition_relationship.name) (product_definition_relationship.name = 'sequence') (product_definition_relationship.name = 'derivation')}
item_solution_ relationship to multi_language_ string (as description)	PATH		144	product_definition_relationship {[product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition => solution_based_product_definition} multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
item_solution_ relationship to item_solution (as related)	PATH			product_definition_relationship product_definition_relationship.related_product_definition -> product_definition => solution_based_product_definition
item_solution_ relationship to item_solution (as relating)	PATH			product_definition_relationship product_definition_relationship.relating_product_definition -> product_definition => solution_based_product_definition
LOT_ CONFIGURATION	lot_effectivity	41		{lot_effectivity <= effectivity <- [effectivity_assignment.assigned_effectivity effectivity_assignment => effective_context_assignment => applied_effective_context_assignment] [effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment => applied_effective_item_assignment]}

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
lot_id	lot_effectivity. effectivity_lot_id	41		
lot_size	measure_with_unit	41		lot_effectivity lot_effectivity.effectivity_lot_size ->
measure_with_unit
PHYSICAL_ INSTANCE	product_definition	41		{product_definition [product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'physical occurrence']
[product_definition.name = 'physical instance']}
description	product_definition. description	41		
lot_id	product_definition. id	41		
serial_number	product_definition. id	41		
physical_ instance to multi_language_ string (as description)	PATH			product_definition {product_definition [product_definition.frame_of_reference ->
product_definition_context <=
application_context_element
application_context_element.name = 'physical occurrence']
[product_definition.name = 'physical instance']}
multi_language_attribute_item = product_definition
multi_language_attribute_item <-
applied_multi_language_attribute_assignment.items[1]
applied_multi_language_attribute_assignment
{applied_multi_language_attribute_assignment <=
multi_language_attribute_assignment
multi_language_attribute_assignment.attribute_name = 'description'}
physical_ instance to item_ version (as is_based_on)	PATH			product_definition product_definition.formation ->
product_definition_formation

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
physical_ instance to product_ specification (as is_based_on)	PATH			product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']} configuration_design_item = product_definition configuration_design_item <- configuration_design.design configuration_design {configuration_design.name = 'physical instance basis'} configuration_design.configuration -> configuration_item => {configuration_item.description = 'product specification'} configurable_item
PHYSICAL_ INSTANCE_USAGE	assembly_component_ usage	44		{assembly_component_usage <= product_definition_usage <= product_definition_relationship product_definition_relationship.name = 'physical occurrence usage'}
physical_ instance_usage to physical_ instance (as physical_ assembly)	PATH			assembly_component_usage <= product_definition_usage <= product_definition_relationship {product_definition_relationship.name = 'physical occurrence usage'} product_definition_relationship.relating_product_definition -> product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
physical_ instance_usage to physical_ instance (as physical_ component)	PATH			assembly_component_usage <= product_definition_usage <= product_definition_relationship {product_definition_relationship.name = 'physical occurrence usage'} product_definition_relationship.related_product_definition -> product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']}
physical_ instance_usage to item_instance (as role)	PATH			assembly_component_usage <= product_definition_usage <= product_definition_relationship {product_definition_relationship.name = 'physical occurrence usage'} product_definition_relationship.related_product_definition -> product_definition <- {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'physical occurrence'] [product_definition.name = 'physical instance']} product_definition_relationship.related_product_definition product_definition_relationship {product_definition_relationship.name = 'physical realization'} product_definition_relationship.relating_product_definition -> product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [(product_definition.name = 'quantified instance') (product_definition.name = 'selected instance') (product_definition.name = 'single instance')]}

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
PROCESS_ OPERATION	action	41		{action action.description = 'process operation'}
description	<action_method. purpose> <action_method. consequence>	41 41		action {action.description = 'process operation'} action.chosen_method -> action_method <action_method.purpose> <action_method.consequence>
id	action.id	41		
name	action.name	41		
process_type	action_method. description	41		action {action.description = 'process operation'} action.chosen_method -> action_method action_method.description
process_ operation to multi_language_ string (as description)	PATH		81	action {action.description = 'process operation'} action.chosen_method -> action_method multi_language_attribute_item = action_method multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment (multi_language_attribute_assignment.attribute_name = 'purpose') (multi_language_attribute_assignment.attribute_name = 'consequence')}
process_ operation to cartesian_ coordinate_space (as is_defined_ in)	PATH			action {action.description = 'process operation'} characterized_action_definition = action characterized_action_definition <- action_property.definition action_property <- action_property_representation.property action_property_representation action_property_representation.representation ->

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation {[representation.name = 'reference system'] [representation representation.items[1] -> representation_item => geometric_representation_item => placement => axis2_placement_3d]} representation.context_of_items -> representation_context => geometric_representation_context
process_ operation to organization (as location)	PATH			action {action.description = 'process operation'} organization_item = action organization_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment.role -> organization_role organization_role.name = 'location'} organization_assignment.assigned_organization -> organization
process_ operation to multi_language_ string (as name)	PATH		79	action {action.description = 'process operation'} multi_language_attribute_item = action multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
process_ operation to property_value (as process_ duration)	PATH			action {action.description = 'process operation'} characterized_action_definition = action characterized_action_definition <- action_property.definition action_property <- action_property_representation.property action_property_representation action_property_representation.representation -> representation => {representation.name = 'duration'} property_value_representation
PRODUCT_CLASS	product_class	214		product_class <= [characterized_object] [product_concept]
description	product_concept. description	44		product_class <= product_concept product_concept.description
id	product_concept.id	44		product_class <= product_concept product_concept.id
level_type	characterized_object. name	41		product_class <= characterized_object characterized_object.name
name	product_concept.name	44		product_class <= product_concept product_concept.name
version_id	identification_ assignment.assigned_ id		176,33	product_class version_item = product_class version_item <- version_assignment.items[1] version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'}

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				identification_assignment.assigned_id
product_class to multi_language_string (as description)	PATH		135	product_class multi_language_attribute_item = product_class multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'product_concept.description'}
product_class to multi_language_string (as name)	PATH		135	product_class multi_language_attribute_item = product_class multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'product_concept.name'}
product_class to product_component (as root)	PATH			product_class <= product_concept <- configuration_item.item_concept configuration_item => configuration_item_with_context => product_component
PRODUCT_CLASS_HIERARCHY	product_concept_relationship	44		{product_concept_relationship.name = 'class hierarchy'}
product_class_hierarchy to product_class (as sub_class)	PATH			product_concept_relationship product_concept_relationship.related_product_concept -> product_concept => product_class
product_class_hierarchy to product_class (as super_class)	PATH			product_concept_relationship product_concept_relationship.relating_product_concept -> product_concept => product_class

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
PRODUCT_ COMPONENT	product_component	214		product_component <= [configuration_item_with_context] [characterized_object]
description	configuration_item. description	44		product_component <= configuration_item_with_context <= configuration_item configuration_item.description
id	configuration_item. id	44		product_component <= configuration_item_with_context <= configuration_item configuration_item.id
instance_ required	configuration_item. purpose	44		product_component <= configuration_item_with_context <= configuration_item configuration_item.purpose {(configuration_item.purpose = 'instance required') (configuration_item.purpose = 'no instance required')}
name	configuration_item. name	44		product_component <= configuration_item_with_context <= configuration_item configuration_item.name
version_id	identification_ assignment.assigned_ id		177,33	product_component version_item = product_component version_item <- version_assignment.items[1] version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
product_ component to multi_language_ string (as description)	PATH		136	product_component multi_language_attribute_item = product_component multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'configuration_item.description'}
product_ component to class_category_ association (as is_ influenced_by)	PATH		32	product_component group_item = product_component group_item <- applied_group_assignment.items[i] applied_group_assignment <= group_assignment {group_assignment.role -> group_role group_role.name = 'class usage influence'} group_assignment.assigned_group -> group <- {group.name = 'influencing category group'} group_assignment.assigned_group group_assignment => {group_assignment.role -> group_role group_role.name = 'class usage group member'} group_member_assignment group_member_assignment.items[i] -> group_member_item group_member_item = product_concept_feature_category_usage product_concept_feature_category_usage {product_concept_feature_category_usage <= group_assignment group_assignment.role -> group_role (group_role.name = 'mandatory category usage')

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				(group_role.name = 'optional category usage')}
product_ component to application_ context (as is_relevant_ for)	PATH			product_component <= configuration_item_with_context configuration_item_with_context.frame_of_reference -> (configuration_item_context) (configuration_item_context <= application_context_element <- application_context_element_relationship.relating_context_element application_context_element_relationship {application_context_element_relationship *} (application_context_element_relationship.related_context_element ->) (application_context_element_relationship.relating_context_element ->) application_context_element => configuration_item_context)
product_ component to multi_language_ string (as name)	PATH		136	product_component multi_language_attribute_item = product_component multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'configuration_item.name'}
PRODUCT_ COMPONENT_ RELATIONSHIP	configuration_item_ relationship	44		{configuration_item_relationship [configuration_item_relationship.relating_configuration_item ->] [configuration_item_relationship.related_configuration_item ->] configuration_item => configuration_item_with_context => product_component}
description	configuration_item_ relationship. description	44		
relation_type	configuration_item_ relationship.name	44		{(configuration_item_relationship.name) (configuration_item_relationship.name = 'decomposition')}

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
product_ component_ relationship to multi_language_ string (as description)	PATH		97	configuration_item_relationship multi_language_attribute_item = configuration_item_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
product_ component_ relationship to product_ component (as related)	PATH			configuration_item_relationship configuration_item_relationship.related_configuration_item -> configuration_item => configuration_item_with_context => product_component
product_ component_ relationship to product_ component (as relating)	PATH			configuration_item_relationship configuration_item_relationship.relating_configuration_item -> configuration_item => configuration_item_with_context => product_component
PRODUCT_FUNCTION	product_definition	41	60	{product_definition [product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'] [product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product product.frame_of_reference[i] -> product_context <= application_context_element application_context_element.name = 'functionality']}
description	product_definition. description	41		

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
id	product_definition. id	41		
name	product_definition. name	41		
version_id	identification_ assignment.assigned_ id	41	179,33	product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'} version_item = product_definition version_item <- version_assignment.items[1] version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
product_function to multi_ language_string (as description)	PATH			product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'} multi_language_attribute_item = product_definition multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
product_function to application_ context (as is_relevant_ for)	PATH			product_definition product_definition.frame_of_reference -> (product_definition_context {product_definition_context <= application_context_element

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				application_context_element.name = 'functional definition'}) (product_definition_context <= application_context_element <- {application_context_element.name = 'functional definition'} application_context_element_relationship.relating_context_element application_context_element_relationship {application_context_element_relationship *} (application_context_element_relationship.related_context_element ->) (application_context_element_relationship.relating_context_element ->) application_context_element => {application_context_element.name = 'functional definition'} product_definition_context)
product_function to multi_ language_string (as name)	PATH			product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'} multi_language_attribute_item = product_definition multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
PRODUCT_FUNCTION_ RELATIONSHIP	product_definition_ relationship	41		{product_definition_relationship [product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'}
description	product_definition_ relationship. description	41		

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
relation_type	product_definition_ relationship.name	41		{(product_definition_relationship.name) (product_definition_relationship.name = 'specialization') (product_definition_relationship.name = 'version sequence')}
product_function_ relationship to multi_language_ string (as description)	PATH		144	product_definition_relationship {product_definition_relationship [product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'} multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
product_function_ relationship to product_function (as related)	PATH			product_definition_relationship product_definition_relationship.related_product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'}
product_function_ relationship to product_function (as relating)	PATH			product_definition_relationship product_definition_relationship.relating_product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'functional definition'}
PRODUCT_ SPECIFICATION	configurable_item	44		configurable_item <= configuration_item {configuration_item.description = 'product specification'}

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
name	configuration_item. name	44		configurable_item <= configuration_item configuration_item.name
version_id	identification_ assignment.assigned_ id	41	169,33	configurable_item {configurable_item <= configuration_item configuration_item.description = 'product specification'} version_item = configurable_item version_item <- version_assignment.items[1] version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
product_ specification to item_version (as explicit_ product_version)	PATH			configurable_item <= configuration_item <- {configuration_item.description = 'product specification'} configuration_design.configuration configuration_design {configuration_design.name = 'explicit product version'} configuration_design.design -> configuration_design_item configuration_design_item = product_definition_formation product_definition_formation
product_ specification to multi_language_ string (as name)	PATH		95	configurable_item {configurable_item <= configuration_item configuration_item.description = 'product specification'} multi_language_attribute_item = configurable_item multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				multi_language_attribute_assignment.attribute_name = \ 'configuration_item.name'}
product_ specification to product_class (as product_ class)	PATH		167	configurable_item <= configuration_item {configuration_item.description = 'product specification'} configuration_item.item_concept -> product_concept => product_class
product_ specification to specification (as specification)	PATH		167	configurable_item {configurable_item <= configuration_item configuration_item.description = 'product specification'} configurable_item.item_concept_feature[i] -> product_concept_feature_association product_concept_feature_association.feature -> (product_concept_feature) (product_concept_feature => package_product_concept_feature)
REPLACED_ DEFINITION_ RELATIONSHIP	product_definition_ relationship	41		{product_definition_relationship {product_definition_relationship.name = 'definition replacement'} [product_definition_relationship.relating_product_definition ->] [product_definition_relationship.related_product_definition ->] product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
description	product_definition_ relationship. description	41		

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
replaced_ definition_ relationship to multi_language_ string (as description)	PATH		144	product_definition_relationship {product_definition_relationship.name = 'definition replacement'} multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
REPLACED_USAGE_ RELATIONSHIP	product_definition_ substitute	41		{product_definition_substitute [product_definition_substitute.context_relationship -> product_definition_relationship product_definition_relationship.related_product_definition ->] [product_definition_substitute.substitute_definition ->] product_definition => product_definition_occurrence}
description	product_definition_ substitute. description	41		
replaced_usage_ relationship to multi_language_ string (as description)	PATH		146	product_definition_substitute multi_language_attribute_item = product_definition_substitute multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
SERIAL_ CONFIGURATION	serial_numbered_ effectivity	41		{serial_numbered_effectivity <= effectivity <- [effectivity_assignment.assigned_effectivity effectivity_assignment => effective_context_assignment => applied_effective_context_assignment] [effectivity_assignment.assigned_effectivity effectivity_assignment => effective_item_assignment =>

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_effective_item_assignment]}
serial_end_ number	serial_numbered_ effectivity. effectivity_end_id	41		
serial_start_ number	serial_numbered_ effectivity. effectivity_start_id	41		
SOLUTION_ CERTIFICATION	certification	41		
certification_ type	certification_type. description	41		certification certification.kind -> certification_type certification_type.description
name	certification.name	41		
purpose	certification. purpose	41		
solution_ certification to date_time (as certification_ date)	PATH			certification (date_item = certification date_item <- applied_date_assignment.items[i] applied_date_assignment <= date_assignment date_assignment.assigned_date -> date => {date <- date_assignment.assigned_date date_assignment date_assignment.role -> date_role date_role.name = 'certification date'} calendar_date) (date_and_time_item = certification date_and_time_item <- applied_date_and_time_assignment.items[i]

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				applied_date_and_time_assignment <= date_and_time_assignment date_and_time_assignment.assigned_date_and_time -> date_and_time {date_and_time <- date_and_time_assignment.assigned_date_and_time date_and_time_assignment date_and_time_assignment.role -> date_time_role date_time_role.name = 'certification date'})
solution_ certification to item_solution (as certified_ solution)	PATH		17	certification <- certification_assignment.assigned_certification certification_assignment => {certification_assignment.role -> certification_role certification_role.name = 'solution certification'} applied_certification_assignment applied_certification_assignment.items[i] -> certification_item certification_item = solution_based_product_definition solution_based_product_definition
solution_ certification to multi_language_ string (as name)	PATH		92	certification multi_language_attribute_item = certification multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
solution_ certification to multi_language_ string (as purpose)	PATH		92	certification multi_language_attribute_item = certification multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				multi_language_attribute_assignment.attribute_name = 'purpose'}
SOLUTION_ INSTANCE_ ASSOCIATION	product_definition_ relationship	41		{product_definition_relationship product_definition_relationship.name = 'solution instance association'}
solution_ instance_ association to item_instance (as instance)	PATH			product_definition_relationship {product_definition_relationship.name = 'solution instance association'} product_definition_relationship.relating_product_definition -> product_definition => {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'} product_definition_occurrence
solution_ instance_ association to item_solution (as solution)	PATH			product_definition_relationship {product_definition_relationship.name = 'solution instance association'} product_definition_relationship.relating_product_definition -> product_definition => solution_based_product_definition
SPECIFICATION	(product_concept_ feature) (package_product_ concept_feature)	44 214		{package_product_concept_feature <= product_concept_feature}
description	product_concept_ feature.description	44		package_product_concept_feature <= product_concept_feature product_concept_feature.description
id	product_concept_ feature.id	44		package_product_concept_feature <= product_concept_feature product_concept_feature.id
name	product_concept_ feature.name	44		package_product_concept_feature <= product_concept_feature product_concept_feature.name
package	IDENTICAL MAPPING			
version_id	identification_ assignment.assigned_ id		178,33	product_concept_feature version_item = product_concept_feature version_item <- version_assignment.items[1]

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
specification to specification_ category (as category)	PATH		32	(product_concept_feature) (package_product_concept_feature <= product_concept_feature) group_member_item = product_concept_feature group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment {group_assignment.role -> group_role group_role.name = 'specification category member'} group_assignment.assigned_group -> (group => product_concept_feature_category) (group => product_concept_feature_category => exclusive_product_concept_feature_category)
specification to multi_language_ string (as description)	PATH		137	product_concept_feature multi_language_attribute_item = product_concept_feature multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
specification to multi_language_ string (as name)	PATH		137	product_concept_feature multi_language_attribute_item = product_concept_feature multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				{applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
SPECIFICATION_ CATEGORY	(product_concept_ feature_category) (exclusive_product_ concept_feature_ category)	214 214		{(product_concept_feature_category <=) (exclusive_product_concept_feature_category <= product_concept_feature_category <=) group}
description	group.description	41		(product_concept_feature_category <=) (exclusive_product_concept_feature_category <= product_concept_feature_category <=) group group.description
id	group.name	41		(product_concept_feature_category <=) (exclusive_product_concept_feature_category <= product_concept_feature_category <=) group group.name
implicit_ exclusive_ condition	IDENTICAL MAPPING			
specification_ category to multi_language_ string (as description)	PATH		139	(product_concept_feature_category) (exclusive_product_concept_feature_category <= product_concept_feature_category) multi_language_attribute_item = product_concept_feature_category multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'group.description'}
SPECIFICATION_ EXPRESSION	conditional_concept_ feature	44		{conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'expression'}
description	product_concept_ feature.description	44		conditional_concept_feature <= product_concept_feature

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_concept_feature.description
operation	concept_feature_ relationship_with_ condition. conditional_operator	44		conditional_concept_feature conditional_concept_feature.condition -> concept_feature_relationship_with_condition concept_feature_relationship_with_condition.conditional_operator {(concept_feature_relationship_with_condition.conditional_operator= 'AND') (concept_feature_relationship_with_condition.conditional_operator= 'OR') (concept_feature_relationship_with_condition.conditional_operator= 'ONEOF') (concept_feature_relationship_with_condition.conditional_operator= 'NOT')}
specification_ expression to multi_language_ string (as description)	PATH		137	conditional_concept_feature <= product_concept_feature multi_language_attribute_item = product_concept_feature multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
specification_ expression to specification (as operand)	PATH			conditional_concept_feature conditional_concept_feature.condition -> concept_feature_relationship_with_condition <= concept_feature_relationship < concept_feature_relationship.relating_product_concept_feature -> > < concept_feature_relationship.related_product_concept_feature -> > (product_concept_feature) (product_concept_feature => package_product_concept_feature)
specification_ expression to specification_ expression (as operand)	PATH			conditional_concept_feature conditional_concept_feature.condition -> concept_feature_relationship_with_condition <= concept_feature_relationship < concept_feature_relationship.relating_product_concept_feature -> > < concept_feature_relationship.related_product_concept_feature -> > product_concept_feature => conditional_concept_feature

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
SPECIFICATION_ INCLUSION	conditional_concept_ feature	44		{[conditional_concept_feature <= product_concept_feature product_concept_feature.name = 'inclusion'] [conditional_concept_feature conditional_concept_feature.condition -> concept_feature_relationship_with_condition concept_feature_relationship_with_condition.conditional_operator = 'implication']}
description	product_concept_ feature.description	44		conditional_concept_feature <= product_concept_feature product_concept_feature.description
specification_ inclusion to multi_language_ string (as description)	PATH		137	conditional_concept_feature <= product_concept_feature multi_language_attribute_item = product_concept_feature multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
specification_ inclusion to specification (as if_ condition)	PATH			conditional_concept_feature conditional_concept_feature.condition -> concept_feature_relationship_with_condition <= concept_feature_relationship concept_feature_relationship.relating_product_concept_feature -> (|product_concept_feature|) (product_concept_feature => package_product_concept_feature)
specification_ inclusion to specification_ expression (as if_ condition)	PATH			conditional_concept_feature conditional_concept_feature.condition -> concept_feature_relationship_with_condition <= concept_feature_relationship concept_feature_relationship.relating_product_concept_feature -> product_concept_feature => {product_concept_feature.name = 'expression'} conditional_concept_feature

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
specification_ inclusion to specification (as included_ specification)	PATH			conditional_concept_feature conditional_concept_feature.condition -> concept_feature_relationship_with_condition <= concept_feature_relationship concept_feature_relationship.related_product_concept_feature -> (|product_concept_feature|) (product_concept_feature => package_product_concept_feature)
specification_ inclusion to specification_ expression (as included_ specification)	PATH			conditional_concept_feature conditional_concept_feature.condition -> concept_feature_relationship_with_condition <= concept_feature_relationship concept_feature_relationship.related_product_concept_feature -> product_concept_feature => {product_concept_feature.name = 'expression'} conditional_concept_feature
SPECIFIED_HIGHER_ USAGE	specified_higher_ usage_occurrence	44		
specified_higher_ usage to next_ higher_assembly (as next_usage)	PATH			specified_higher_usage_occurrence specified_higher_usage_occurrence.next_usage -> next_assembly_usage_occurrence
specified_higher_ usage to assembly_ component_ relationship (as upper_usage)	PATH			specified_higher_usage_occurrence specified_higher_usage_occurrence.upper_usage -> assembly_component_usage
SUPPLIER_ SOLUTION	solution_based_ product_definition	214		solution_based_product_definition <= product_definition {(product_definition.name = 'supplier') (product_definition.name = 'technical supplier') (product_definition.name = 'supplier final') (product_definition.name = 'technical supplier final')}
probability_rate	measure_ representation_item	45		solution_based_product_definition <= product_definition

Table 29 – Mapping table specification_control UoF (S7) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation {representation.description = 'supplier probability'} representation.items[i] -> representation_item => {representation_item.name = 'probability rate'} measure_representation_item {measure_representation_item <= measure_with_unit => ratio_measure_with_unit}
supplier	organization.name	41		solution_based_product_definition <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment.role -> organization_role organization_role.name = 'supplier'} organization_assignment.assigned_organization -> organization organization.name
TECHNICAL_ SOLUTION	solution_based_ product_definition	214		solution_based_product_definition <= product_definition {(product_definition.name = 'technical') (product_definition.name = 'technical supplier') (product_definition.name = 'technical final') (product_definition.name = 'technical supplier final')}

Table 29 – Mapping table specification_control UoF (S7) (concluded)

Application element	AIM element	Source	Rules	Reference path
description	product_definition. description	41		solution_based_product_definition <= product_definition product_definition.description
technical_ solution to multi_language_ string (as description)	PATH		163	solution_based_product_definition multi_language_attribute_item = solution_based_product_definition multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'product_definition.description'}

Table 30 – Mapping table process_plan UoF (S8)

Application element	AIM element	Source	Rules	Reference path
EXCLUSIVE_ PROCESS_ OPERATION	action	41		{action action.description = 'exclusive process operation'}
exclusive_ process_ operation to process_ operation (as element)	PATH			action <- {action.description = 'exclusive process operation'} action_relationship.relating_action action_relationship {action_relationship.name = 'exclusive process operation membership'} action_relationship.related_action ->< br>action {action.description = 'process operation'}
GENERAL_PROCESS_ PROPERTY	action_property	49		
description	action_property. description	49		
property_type	action_property.name	49		
general_process_ property to multi_language_ string (as description)	PATH		82	action_property multi_language_attribute_item = action_property multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
general_process_ property to external_model (as property_ value)	PATH			action_property <- action_property_representation.property action_property_representation action_property_representation.representation ->< br>representation => externally_defined_representation

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
general_process_property to geometric_model (as property_value)	PATH			action_property <- action_property_representation.property action_property_representation action_property_representation.representation -> representation => shape_representation
general_process_property to property_value (as property_value)	PATH			action_property <- action_property_representation.property action_property_representation action_property_representation.representation -> representation => property_value_representation
general_process_property to styled_model (as property_value)	PATH			action_property <- action_property_representation.property action_property_representation action_property_representation.representation -> representation => (draughting_model) (mechanical_design_geometric_presentation_representation)
MAKE_FROM_RELATIONSHIP	make_from_usage_option	44		
description	product_definition_relationship.description	41		make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.description
name	product_definition_relationship.name	41		make_from_usage_option <= product_definition_usage <= product_definition_relationship product_definition_relationship.name
make_from_relationship to multi_language_string (as description)	PATH		118	make_from_usage_option multi_language_attribute_item = make_from_usage_option multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				multi_language_attribute_assignment.attribute_name = \ 'product_definition_relationship.description'}
make_from_ relationship to multi_language_ string (as name)	PATH		118	make_from_usage_option multi_language_attribute_item = make_from_usage_option multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'product_definition_relationship.name'}
MATED_ITEM_ ASSOCIATION	product_definition_ relationship	41		{product_definition_relationship [product_definition_relationship.name = 'mating membership'] [product_definition_relationship.relating_product_definition ->] product_definition group_member_item = product_definition group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment {group_assignment.role ->] group_role group_role.name = 'mating'} group_assignment.assigned_group ->] group group.name = 'mating definition'] [product_definition_relationship.related_product_definition ->] product_definition =>] {product_definition.frame_of_reference ->] product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'} product_definition_occurrence]}

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
mated_item_ association to mating_ definition (as relating)	PATH			product_definition_relationship {product_definition_relationship.name = \ 'mating membership'} product_definition_relationship.relating_product_definition ->product_definition {product_definition group_member_item = product_definition group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment [group_assignment.role ->group_role group_role.name = 'mating'] [group_assignment.assigned_group ->group group.name = 'mating definition']}
MATED_ITEM_ RELATIONSHIP	mated_item_ relationship	214		mated_item_relationship <= [product_definition_relationship] [product_definition {product_definition <- product_definition_relationship.related_product_definition product_definition_relationship {product_definition_relationship.name = 'mating context'} product_definition_relationship.relating_product_definition ->product_definition group_member_item = product_definition group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment [group_assignment.role ->group_role group_role.name = 'mating'] [group_assignment.assigned_group ->group

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				group.name = 'mating definition']}]
mated_item_ relationship to shape_aspect_ relationship (as mated_shape)	PATH			mated_item_relationship <= product_definition_relationship [product_definition_relationship.relating_product_definition ->] product_definition {product_definition => product_definition_occurrence} characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- shape_aspect_relationship.relating_shape_aspect] [product_definition_relationship.related_product_definition ->] product_definition {product_definition => product_definition_occurrence} characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape shape_aspect <- shape_aspect_relationship.related_shape_aspect] shape_aspect_relationship

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
mated_item_ relationship to quantified_ instance (as mating_ material)	PATH			mated_item_relationship <= product_definition <- product_definition_relationship.relating_product_definition product_definition_relationship {product_definition_relationship.name = 'mating material'} product_definition_relationship.related_product_definition -> product_definition => {product_definition.name = 'quantified instance'} product_definition_occurrence
mated_item_ relationship to mated_item_ association (as related)	PATH			mated_item_relationship <= product_definition_relationship product_definition_relationship.related_product_definition -> product_definition <- {[product_definition.name = 'quantified instance'] [product_definition => product_definition_occurrence]} product_definition_relationship.related_product_definition product_definition_relationship {product_definition_relationship.name = 'mating membership'}
mated_item_ relationship to mated_item_ association (as relating)	PATH			mated_item_relationship <= product_definition_relationship product_definition_relationship.relating_product_definition -> product_definition <- {[product_definition.name = 'quantified instance'] [product_definition => product_definition_occurrence]} product_definition_relationship.related_product_definition product_definition_relationship {product_definition_relationship.name = 'mating membership'}

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
mated_item_ relationship to geometric_model_ relationship_ with_ transformation (as transformation)	PATH			mated_item_relationship <= product_definition_relationship [product_definition_relationship.relating_product_definition -> product_definition {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence'] [product_definition => product_definition_occurrence]} characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <- {representation => shape_representation} representation_relationship.rep_2] [product_definition_relationship.related_product_definition -> product_definition <- {[product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional occurrence']

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				[product_definition => product_definition_occurrence]} characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <- {representation => shape_representation} representation_relationship.rep_1] representation_relationship => representation_relationship_with_transformation
mated_item_ relationship to template_ instance (as transformation)	PATH			mated_item_relationship <= product_definition_relationship [product_definition_relationship.relating_product_definition -> product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item =>] [product_definition_relationship.related_product_definition -> product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <- {representation => shape_representation} representation_map.mapped_representation representation_map <- mapped_item.mapping_source] mapped_item

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				{mapped_item.mapping_target -> representation_item => geometric_representation_item => (placement) (cartesian_transformation_operator cartesian_transformation_operator.scl = 1.0)}
MATING_ DEFINITION	product_definition	41	32	{[product_definition product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'] [product_definition <- product_definition_relationship.relating_product_definition product_definition_relationship product_definition_relationship.name = 'mating membership'] [product_definition group_member_item = product_definition group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment [group_assignment.role -> group_role group_role.name = 'mating'] [group_assignment.assigned_group -> group group.name = 'mating definition']]}
mating_type	product_definition. description	41		{product_definition (product_definition.description) (product_definition.description = 'welding') (product_definition.description = 'laser-welding') (product_definition.description = 'flanging') (product_definition.description = 'clinching') (product_definition.description = 'clipping')

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				(product_definition.description = 'clasping') (product_definition.description = 'glueing') (product_definition.description = 'brazing') (product_definition.description = 'sewing') (product_definition.description = 'riveting') (product_definition.description = 'spot welding') (product_definition.description = 'foam injection') (product_definition.description = 'bolted joint') (product_definition.description = 'doweling') (product_definition.description = 'press fit') (product_definition.description = 'stapling')}
PROCESS_COST_ PROPERTY	action_property	49		{action_property action_property.name = 'cost property'}
process_cost_ property to property_value (as cost_ property_value)	PATH			action_property <- {action_property.name = 'cost property'} action_property_representation.property action_property_representation action_property_representation.representation -> representation => property_value_representation
PROCESS_ OPERATION	action	41		{action action.description = 'process operation'}
description	<action_method. purpose> <action_method. consequence>	41 41		action {action.description = 'process operation'} action.chosen_method -> action_method <action_method.purpose> <action_method.consequence>
id	action.id	41		
name	action.name	41		
process_type	action_method. description	41		action {action.description = 'process operation'} action.chosen_method -> action_method action_method.description

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
process_ operation to multi_language_ string (as description)	PATH		81	action {action.description = 'process operation'} action.chosen_method -> action_method multi_language_attribute_item = action_method multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment (multi_language_attribute_assignment.attribute_name = 'purpose') (multi_language_attribute_assignment.attribute_name = 'consequence')}
process_ operation to cartesian_ coordinate_space (as is_defined_ in)	PATH			action {action.description = 'process operation'} characterized_action_definition = action characterized_action_definition <- action_property.definition action_property <- action_property_representation.property action_property_representation action_property_representation.representation -> representation {[representation.name = 'reference system'] [representation representation.items[1] -> representation_item => geometric_representation_item => placement => axis2_placement_3d]} representation.context_of_items -> representation_context => geometric_representation_context

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
process_ operation to organization (as location)	PATH			action {action.description = 'process operation'} organization_item = action organization_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment.role -> organization_role organization_role.name = 'location'} organization_assignment.assigned_organization -> organization
process_ operation to multi_language_ string (as name)	PATH		79	action {action.description = 'process operation'} multi_language_attribute_item = action multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
process_ operation to property_value (as process_ duration)	PATH			action {action.description = 'process operation'} characterized_action_definition = action characterized_action_definition <- action_property.definition action_property <- action_property_representation.property action_property_representation action_property_representation.representation -> representation => {representation.name = 'duration'} property_value_representation

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
PROCESS_ OPERATION_INPUT_ OR_OUTPUT	process_product_ association	49		
description	process_product_ association. description	49		
quantity	measure_with_unit	41	13	process_product_association [(process_product_association.defined_product -> characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship {(product_definition_relationship => product_definition_usage => assembly_component_usage) (product_definition_relationship.name = 'mating membership')} characterized_product_definition = product_definition_relationship characterized_definition = characterized_product_definition characterized_definition <-) (process_product_association.defined_product -> characterized_product_definition characterized_product_definition = product_definition product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'} characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <-) property_definition.definition property_definition represented_definition = property_definition represented_definition <-

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.definition property_definition_representation {property_definition_representation.name = 'process operation quantity'} property_definition_representation.used_representation ->] [[process_product_association.process -> product_definition_process => action <- {action.description = 'process operation'} action_relationship.related_action] [process_plan_item = process_product_association process_plan_item <- process_plan_assignment.items[i] process_plan_assignment <= action_assignment {action_assignment.role -> action_role action_role.name = 'process plan context'} action_assignment.assigned_action -> action <- {action => product_definition_process => process_plan} action_relationship.relating_action] action_relationship {action_relationship.name = 'process plan operation assignment'} characterized_action_definition = action_relationship characterized_action_definition <- action_property.definition action_property <- action_property_representation.property action_property_representation {action_property_representation.name = 'process operation quantity'} action_property_representation.representation ->]

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				representation {(representation.description = 'input quantity') (representation.description = 'output quantity')} representation.items[1] -> representation_item => {representation_item.name = 'quantity'} measure_representation_item <= measure_with_unit
role	process_product_association.name	49		{(process_product_association.name = 'input') (process_product_association.name = 'output')}
process_operation_input_or_output to shape_aspect (as concerned_shape)	PATH			process_product_association process_product_association.process -> product_definition_process <= action => property_process <- process_property_association.process process_property_association {process_property_association.name = 'concerned shape aspect'} process_property_association.property_or_shape -> property_or_shape_select property_or_shape_select = shape_definition shape_definition = shape_aspect shape_aspect
process_operation_input_or_output to multi_language_string (as description)	PATH		132	process_product_association multi_language_attribute_item = process_product_association multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
process_operation_input_or_output to assembly_component_relationship (as element)	PATH			process_product_association process_product_association.defined_product ->characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship => product_definition_usage => assembly_component_usage
process_operation_input_or_output to design_discipline_item_definition (as element)	PATH			process_product_association process_product_association.defined_product ->characterized_product_definition characterized_product_definition = product_definition product_definition {product_definition.frame_of_reference ->product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
process_operation_input_or_output to item_instance (as element)	PATH			process_product_association process_product_association.defined_product ->characterized_product_definition (characterized_product_definition = product_definition product_definition => product_definition_occurrence) (characterized_product_definition = product_definition_relationship product_definition_relationship => {(product_definition_relationship.name = 'quantified instance usage') (product_definition_relationship.name = 'selected instance usage') (product_definition_relationship.name = 'single instance usage')} product_definition_usage => assembly_component_usage)

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
process_ operation_input_ or_output to mated_item_ association (as element)	PATH			process_product_association process_product_association.defined_product -> characterized_product_definition characterized_product_definition = product_definition_relationship product_definition_relationship {[product_definition_relationship.name = 'mating membership'] [product_definition_relationship.relating_product_definition ->] product_definition group_member_item = product_definition group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment [group_assignment.role ->] group_role group_role.name = 'mating'] [group_assignment.assigned_group ->] group group.name = 'mating definition']]}
process_ operation_input_ or_output to process_plan_ operation_ assignment (as operation)	PATH		165,13	process_product_association [process_product_association.process ->] product_definition_process =>] action <- {action.description = 'process operation'} action_relationship.related_action] [process_plan_item = process_product_association process_plan_item <- process_plan_assignment.items[i] process_plan_assignment <= action_assignment {action_assignment.role ->] action_role action_role.name = 'process plan context'} action_assignment.assigned_action ->

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				action <- {action => product_definition_process => process_plan} action_relationship.relating_action] action_relationship {action_relationship.name = 'process plan operation assignment'}
process_ operation_input_ or_output to transformation (as transformation)	PATH			[process_product_association process_product_association.process ->> product_definition_process => action {action.description = 'process operation'} characterized_action_definition = action characterized_action_definition <- (action_property.definition action_property <- action_property_representation.property action_property_representation action_property_representation.representation ->> representation <- {representation.name = 'reference system'}) (action_resource_requirement.operations[i] action_resource_requirement => {action_resource_requirement.name = 'reference tool'} requirement_for_action_resource requirement_for_action_resource.resources[1] ->> action_resource => product_definition_resource <= product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <-) representation_relationship.rep_2] [process_product_association process_product_association.defined_product -> characterized_product_definition {(characterized_product_definition = product_definition product_definition {(product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template') (product_definition group_member_item = product_definition group_member_item <- group_member_assignment.items[i] group_member_assignment <= group_assignment [group_assignment.role -> group_role group_role.name = 'mating'] [group_assignment.assigned_group -> group group.name = 'mating definition'])})

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				(characterized_product_definition = product_definition product_definition => product_definition_occurrence) (characterized_product_definition = product_definition_relationship product_definition_relationship => product_definition_usage => assembly_component_usage)} characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <- representation_relationship.rep_1] representation_relationship => representation_relationship_with_transformation representation_relationship_with_transformation.transformation_operator -> transformation (transformation = item_defined_transformation item_defined_transformation) (transformation = functionally_defined_transformation functionally_defined_transformation)

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
PROCESS_ OPERATION_ RELATIONSHIP	action_relationship	41		{action_relationship [action_relationship.related_action ->] [action_relationship.relating_action ->] action action.description = 'process operation'}
condition	relationship_ condition.condition_ description	49		action_relationship relationship_with_condition = action_relationship relationship_with_condition <- relationship_condition.applicable_relationships[i] -> relationship_condition relationship_condition.condition_description
description	action_relationship. description	41		
relation_type	action_relationship. name	41		{(action_relationship.name) (action_relationship.name = 'sequence') (action_relationship.name = 'alternative') (action_relationship.name = 'decomposition')}
process_ operation_ relationship to multi_language_ string (as description)	PATH		83	action_relationship {[action_relationship.related_action ->] [action_relationship.relating_action ->] action action.description = 'process operation'} multi_language_attribute_item = action_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
process_ operation_ relationship to process_ operation (as related)	PATH			action_relationship action_relationship.related_action -> action {action.description = 'process operation'}

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
process_ operation_ relationship to process_ operation (as relating)	PATH			action_relationship action_relationship.relating_action ->>>>

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
PROCESS_ OPERATION_ RESOURCE_ ASSIGNMENT	action_resource_ requirement	49		
reason	action_resource_ requirement. description	49		
reference_tool	action_resource_ requirement.name		73	{action_resource_requirement ([action_resource_requirement.name = 'reference tool'] [action_resource_requirement => requirement_for_action_resource requirement_for_action_resource.resources[1] -> action_resource => product_definition_resource <= product_definition]) (action_resource_requirement.name = 'non reference tool')}
process_ operation_ resource_ assignment to process_ operation (as operation)	PATH			action_resource_requirement action_resource_requirement.operations[i] -> characterized_action_definition characterized_action_definition = action action {action.description = 'process operation'}
process_ operation_ resource_ assignment to property_value (as process_ duration_for_ resource)	PATH			action_resource_requirement characterized_resource_definition = action_resource_requirement characterized_resource_definition <- resource_property.resource resource_property <- resource_property_representation.property resource_property_representation resource_property_representation.representation -> representation => {representation.description = 'process duration for resource'} property_value_representation

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
process_ operation_ resource_ assignment to multi_language_ string (as reason)	PATH		84	action_resource_requirement multi_language_attribute_item = action_resource_requirement multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
process_ operation_ resource_ assignment to design_ discipline_item_ definition (as resource)	PATH			action_resource_requirement => requirement_for_action_resource requirement_for_action_resource.resources[i] -> action_resource => product_definition_resource <= product_definition
process_ operation_ resource_ assignment to process_ operation_ resource (as resource)	PATH			action_resource_requirement action_resource_requirement.kind -> resource_requirement_type
process_ operation_ resource_ assignment to transformation (as transformation)	PATH			[(action_resource_requirement => {action_resource_requirement.name = 'reference tool'} requirement_for_action_resource requirement_for_action_resource.resources[1] -> action_resource => product_definition_resource <= product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				{property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <-) (action_resource_requirement action_resource_requirement.operations[i] -> characterized_action_definition characterized_action_definition = action action {action.description = 'process operation'} characterized_action_definition = action characterized_action_definition <- action_property.definition action_property <- action_property_representation.property action_property_representation action_property_representation.representation -> representation <- {representation.name = 'reference system'}) representation_relationship.rep_2] [action_resource_requirement => {action_resource_requirement.name = 'non reference tool'} requirement_for_action_resource requirement_for_action_resource.resources[i] -> action_resource => product_definition_resource <= product_definition

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <- representation_relationship.rep_1] representation_relationship => representation_relationship_with_transformation representation_relationship_with_transformation.transformation_operator -> transformation (transformation = item_defined_transformation item_defined_transformation) (transformation = functionally_defined_transformation functionally_defined_transformation)
PROCESS_PLAN	process_plan	214		process_plan <= product_definition_process
description	action.description	41		process_plan <= product_definition_process <= action action.description
name	action.name	41		process_plan <= product_definition_process <= action action.name

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
plan_id	action.id	41		process_plan <= product_definition_process <= action action.id
process_plan to multi_language_string (as description)	PATH		131	process_plan multi_language_attribute_item = process_plan multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'action.description'}
process_plan to organization (as location)	PATH			process_plan organization_item = process_plan organization_item <- applied_organization_assignment.items[i] applied_organization_assignment <= organization_assignment {organization_assignment.role -> organization_role organization_role.name = 'location'} organization_assignment.assigned_organization -> organization
process_plan to multi_language_string (as name)	PATH		131	process_plan multi_language_attribute_item = process_plan multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'action.name'}
process_plan to item_version (as produced_output)	PATH			process_plan <= product_definition_process <- process_product_association.process process_product_association {process_product_association.name = 'produced output'}

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				process_product_association.defined_product -> characterized_product_definition characterized_product_definition = product_definition product_definition product_definition.formation -> product_definition_formation
PROCESS_PLAN_ OPERATION_ ASSIGNMENT	action_relationship	41		{action_relationship.name = 'process plan operation assignment'}
process_plan_ operation_ assignment to process_ operation (as operation)	PATH			action_relationship {action_relationship.name = 'process plan operation assignment'} action_relationship.related_action -> action {action.description = 'process operation'}
process_plan_ operation_ assignment to process_plan (as plan)	PATH			action_relationship {action_relationship.name = 'process plan operation assignment'} action_relationship.relating_action -> action => product_definition_process => process_plan
PROCESS_PLAN_ RELATIONSHIP	action_relationship	41		{action_relationship [action_relationship.relating_action ->] [action_relationship.related_action ->] action => product_definition_process => process_plan}
description	action_relationship. description	41		
relation_type	action_relationship. name	41		

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
process_plan_ relationship to multi_language_ string (as description)	PATH		83	action_relationship {[action_relationship.relating_action ->] [action_relationship.related_action ->] action => product_definition_process => process_plan} multi_language_attribute_item = action_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
process_plan_ relationship to process_plan (as related)	PATH			action_relationship action_relationship.related_action ->> action => product_definition_process => process_plan
process_plan_ relationship to process_plan (as relating)	PATH			action_relationship action_relationship.relating_action ->> action => product_definition_process => process_plan
PROCESS_PLAN_ VERSION	process_plan	214	175,33	process_plan <= {process_plan version_item = process_plan version_item <- version_assignment.items[1] version_assignment <= identification_assignment identification_assignment.role ->> identification_role identification_role.name = 'version id'} product_definition_process
version_id	identification_ assignment.assigned_	41	175,33	process_plan version_item = process_plan

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
	id			version_item <- version_assignment.items[1] version_assignment <= identification_assignment {identification_assignment.role -> identification_role identification_role.name = 'version id'} identification_assignment.assigned_id
PROCESS_PROPERTY	action_property	49		
process_property to process_operation (as described_object)	PATH			action_property action_property.definition -> characterized_action_definition characterized_action_definition = action action {action.description = 'process operation'}
process_property to process_plan (as described_object)	PATH			action_property action_property.definition -> characterized_action_definition characterized_action_definition = action action => product_definition_process => process_plan
PROCESS_QUALITY_PROPERTY	action_property	49		{action_property action_property.name = 'quality property'}
description	action_property.description	49		
process_quality_property to multi_language_string (as description)	PATH		82	action_property {action_property.name = 'quality property'} multi_language_attribute_item = action_property multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
process_quality_ property to property_value (as quality_ property_value)	PATH			action_property <- {action_property.name = 'quality property'} action_property_representation.property action_property_representation action_property_representation.representation -> representation => property_value_representation
PROCESS_STATE	product_definition	41		{product_definition.description = 'process state'}
process_state to design_ discipline_item_ definition (as related_item_ definition)	PATH			product_definition <- {product_definition.description = 'process state'} product_definition_relationship.related_product_definition product_definition_relationship {product_definition_relationship.name = 'process state to related item'} product_definition_relationship.relating_product_definition -> product_definition {product_definition.frame_of_reference -> product_definition_context <= application_context_element application_context_element.name = 'definitional template'}
SAME_TIME_ MACHINING_ RELATIONSHIP	product_definition_ relationship	41		{product_definition_relationship product_definition_relationship.name = 'same time machining' }
description	product_definition_ relationship. description	41		
name	product_definition_ relationship.id	41		
same_time_ machining_ relationship to multi_language_ string (as description)	PATH		144	product_definition_relationship {product_definition_relationship.name = 'same time machining' } multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
same_time_ machining_ relationship to multi_language_ string (as name)	PATH			product_definition_relationship {product_definition_relationship.name = 'same time machining'} multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'id'}
same_time_ machining_ relationship to geometric_model_ relationship_ with_ transformation (as transformation)	PATH			product_definition_relationship {product_definition_relationship.name = 'same time machining' } [product_definition_relationship.relating_product_definition -> product_definition {product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category <- {product_category.name = 'tool'} product_category_relationship.sub_category product_category_relationship product_category_relationship.category -> product_category product_category.name = 'item role'} characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <-

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <- representation_relationship.rep_2] [product_definition_relationship.related_product_definition -> product_definition {product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category <- {product_category.name = 'part'} product_category_relationship.sub_category product_category_relationship product_category_relationship.category -> product_category product_category.name = 'item role'} characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation}

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_representation -> representation <- representation_relationship.rep_1] representation_relationship => representation_relationship_with_transformation
same_time_ machining_ relationship to template_ instance (as transformation)	PATH			product_definition_relationship {product_definition_relationship.name = 'same time machining' } [product_definition_relationship.relating_product_definition -> product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item =>] [product_definition_relationship.related_product_definition -> product_definition characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <-

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <- {representation => shape_representation} representation_map.mapped_representation representation_map <- mapped_item.mapping_source] mapped_item {mapped_item.mapping_target -> representation_item => geometric_representation_item => (placement) (cartesian_transformation_operator cartesian_transformation_operator.scl = 1.0)}
SIMULTANEOUS_ PROCESS_ OPERATION	action	41		{action action.description = 'simultaneous process operation'}
cycle_time	measure_ representation_item	45		action {action.description = 'simultaneous process operation'} characterized_action_definition = action characterized_action_definition <- action_property.definition action_property <- action_property_representation.property action_property_representation

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				action_property_representation.representation -> representation {representation.name = 'duration'} representation.items[i] -> representation_item => {representation_item.name = 'cycle time'} measure_representation_item {measure_representation_item <= measure_with_unit => time_measure_with_unit}
simultaneous_ process_ operation to process_ operation (as element)	PATH			action <- {action.description = 'simultaneous process operation'} action_relationship.relating_action action_relationship {action_relationship.name = 'exclusive process operation membership'} action_relationship.related_action -> action {action.description = 'process operation'}
TOOL_PART_ RELATIONSHIP	product_definition_ relationship	41		{product_definition_relationship product_definition_relationship.name = 'tool part relationship' }
used_technology_ description	product_definition_ relationship. description	41		{product_definition_relationship product_definition_relationship.name = 'tool part relationship' }
tool_part_ relationship to geometric_model_ relationship_ with_ transformation (as transformation)	PATH			product_definition_relationship {product_definition_relationship.name = 'tool part relationship' } [product_definition_relationship.relating_product_definition -> product_definition {product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category <- {product_category.name = 'tool'}

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_category_relationship.sub_category product_category_relationship product_category_relationship.category -> product_category product_category.name = 'item role'} characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <- representation_relationship.rep_2] [product_definition_relationship.related_product_definition -> product_definition {product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category <- {product_category.name = 'part'} product_category_relationship.sub_category product_category_relationship product_category_relationship.category -> product_category

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_category.name = 'item role'} characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <- representation_relationship.rep_1] representation_relationship => representation_relationship_with_transformation
tool_part_ relationship to template_ instance (as transformation)	PATH			product_definition_relationship {product_definition_relationship.name = 'tool part relationship' } [product_definition_relationship.relating_product_definition -> product_definition {product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category <- {product_category.name = 'tool'} product_category_relationship.sub_category product_category_relationship product_category_relationship.category ->

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_category product_category.name = 'item role'} characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation {representation => shape_representation} representation.items[i] -> representation_item =>] [product_definition_relationship.related_product_definition -> product_definition {product_definition.formation -> product_definition_formation product_definition_formation.of_product -> product <- product_related_product_category.products[i] product_related_product_category <= product_category <- {product_category.name = 'part'} product_category_relationship.sub_category product_category_relationship product_category_relationship.category -> product_category

Table 30 – Mapping table process_plan UoF (S8) (continued)

Application element	AIM element	Source	Rules	Reference path
				product_category.name = 'item role'} characterized_product_definition = product_definition characterized_definition = characterized_product_definition characterized_definition <- property_definition.definition property_definition {property_definition => product_definition_shape} represented_definition = property_definition represented_definition <- property_definition_representation.definition property_definition_representation {property_definition_representation => shape_definition_representation} property_definition_representation.used_representation -> representation <- {representation => shape_representation} representation_map.mapped_representation representation_map <- mapped_item.mapping_source] mapped_item {mapped_item.mapping_target -> representation_item => geometric_representation_item => (placement) (cartesian_transformation_operator cartesian_transformation_operator.scl = 1.0)}

Table 30 – Mapping table process_plan UoF (S8) (concluded)

Application element	AIM element	Source	Rules	Reference path
tool_part_ relationship to multi_language_ string (as used_ technology_ description)	PATH			product_definition_relationship {product_definition_relationship.name = 'tool part relationship' } multi_language_attribute_item = product_definition_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}

Table 31 – Mapping table dimension_tolerance UoF (T1)

Application element	AIM element	Source	Rules	Reference path
ANGULAR_LOCATION_ DIMENSION	angular_location	47		
major_angle	angular_location. angle_selection	47		
mirror_target	(shape_aspect) (shape_aspect_ relationship)	41 41		angular_location <= dimensional_location <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect ->(shape_aspect) (shape_aspect <- {shape_aspect => extension} shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship {[shape_aspect_relationship shape_aspect_relationship.name = 'mirroring relationship'] [shape_aspect_relationship => shape_aspect_deriving_relationship]})
origin_ determination	(shape_aspect) (shape_aspect_ relationship) (shape_aspect_ relationship)	41 41 41		angular_location <= dimensional_location <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect ->(shape_aspect) (shape_aspect <- {shape_aspect => extension} shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship {[shape_aspect_relationship shape_aspect_relationship.name = 'mirroring relationship'] [shape_aspect_relationship => shape_aspect_deriving_relationship]}) (shape_aspect <- {shape_aspect => perpendicular_to} shape_aspect_relationship.relating_shape_aspect

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship {[shape_aspect_relationship shape_aspect_relationship.name = 'normal relationship'] [shape_aspect_relationship => shape_aspect_deriving_relationship]})
ANGULAR_SIZE_ DIMENSION	angular_size	47		
full	representation_item. name	43		angular_size <= dimensional_size dimensional_characteristic = dimensional_size dimensional_characteristic <- dimensional_characteristic_representation.dimension dimensional_characteristic_representation dimensional_characteristic_representation.representation -> shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item representation_item.name {(representation_item.name = 'full angle') (representation_item.name = 'half angle')}
major_angle	angular_size.angle_ selection	47		
CURVED_DISTANCE_ DIMENSION	dimensional_location_ with_path	47		
with_curve_ direction	shape_aspect_ relationship.name	41		dimensional_location_with_path <= dimensional_location <= shape_aspect_relationship shape_aspect_relationship.name {(shape_aspect_relationship.name = 'true') (shape_aspect_relationship.name = 'false')}

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
curved_distance_ dimension to shape_aspect (as path)	PATH			dimensional_location_with_path dimensional_location_with_path.path ->
shape_aspect
DEFAULT_SETTING_ ASSOCIATION	(property_definition_ representation) (representation_ relationship)	41 43		({property_definition_representation property_definition_representation.name = 'default setting'}) ({representation_relationship representation_relationship.name = 'default setting'})
default_setting_ association to digital_document (as default_ context)	PATH		27	representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_2 ->
representation =>
externally_defined_representation digital_document_reference_item = externally_defined_representation digital_document_reference_item <-
digital_document_reference.items[1] digital_document_reference <=
document_reference {document_reference.role ->
document_role document_role.name = 'external definition'} document_reference.assigned_document ->
document {document.kind ->
document_type document_type.product_data_type = 'digital'}
default_setting_ association to drawing_sheet (as default_ context)	PATH			representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_2 ->
representation =>
presentation_representation =>
presentation_area =>
drawing_sheet_revision

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
default_setting_ association to item_shape (as default_ context)	PATH			property_definition_representation {property_definition_representation.name = 'default setting'} property_definition_representation.definition -> represented_definition = property_definition property_definition => product_definition_shape
default_setting_ association to view_area (as default_ context)	PATH			representation_relationship {representation_relationship.name = 'default setting'} representation_relationship.rep_2 -> representation => presentation_representation => (presentation_area) (presentation_area => mechanical_design_geometric_presentation_area)
default_setting_ association to general_ tolerances (as default_ value)	PATH			property_definition_representation {property_definition_representation.name = 'default setting'} property_definition_representation.used_representation -> representation => default_tolerance_table
DERIVED_GEOMETRY	derived_shape_aspect	47		
role	shape_aspect.name	41		derived_shape_aspect <= shape_aspect shape_aspect.name
derived_geometry to constructive_ geometric_ element (as geometry)	PATH			derived_shape_aspect <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => geometric_representation_item => (placement =>

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
				axis2_placement_2d) (point => cartesian_point) (point => point_on_curve) (curve)
derived_geometry to shape_aspect (as is_derived_ from)	PATH			derived_shape_aspect <= shape_aspect <- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship {shape_aspect_relationship => shape_aspect_deriving_relationship} shape_aspect_relationship.related_shape_aspect -> shape_aspect
DIMENSION_ TOLERANCE	dimensional_ characteristic_ representation	47		
envelope	PATH			dimensional_characteristic_representation dimensional_characteristic_representation.representation -> shape_dimension_representation <= shape_representation <= representation representation.name {representation.name = 'envelope tolerance'}
name	representation. description	43		dimensional_characteristic_representation dimensional_characteristic_representation.representation -> shape_dimension_representation <= shape_representation <= representation representation.description
significant_ digits	precision_qualifier. precision_value	45		dimensional_characteristic_representation dimensional_characteristic_representation.representation -> shape_dimension_representation <= shape_representation <= representation representation.items[i] ->

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
				{representation_item => measure_representation_item} representation_item => qualified_representation_item qualified_representation_item.qualifiers[1] ->> value_qualifier value_qualifier = precision_qualifier precision_qualifier precision_qualifier.precision_value
value_ interpretation	representation_ context.context_type	43		dimensional_characteristic_representation dimensional_characteristic_representation.representation ->> shape_dimension_representation <= shape_representation <= representation representation.context_of_items ->> representation_context representation_context.context_type {(representation_context.context_type) (representation_context.context_type = 'required') (representation_context.context_type= 'calculated') (representation_context.context_type= 'measured') (representation_context.context_type= 'estimated')}
dimension_ tolerance to limits_and_fits (as is_defined_ by)	PATH			dimensional_characteristic_representation dimensional_characteristic_representation.dimension ->> dimensional_characteristic <- plus_minus_tolerance.toleranced_dimension plus_minus_tolerance plus_minus_tolerance.range ->> tolerance_method_definition tolerance_method_definition = limits_and_fits limits_and_fits

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
dimension_ tolerance to plus_minus_ tolerance (as is_defined_ by)	PATH			dimensional_characteristic_representation dimensional_characteristic_representation.dimension -> dimensional_characteristic <- plus_minus_tolerance.toleranced_dimension plus_minus_tolerance plus_minus_tolerance.range -> tolerance_method_definition tolerance_method_definition = tolerance_value tolerance_value
dimension_ tolerance to value_limit (as is_defined_ by)	PATH			dimensional_characteristic_representation dimensional_characteristic_representation.representation -> shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item => measure_representation_item
dimension_ tolerance to value_range (as is_defined_ by)	PATH			dimensional_characteristic_representation dimensional_characteristic_representation.representation -> shape_dimension_representation
dimension_ tolerance to feature_ parameter (as is_ tolerancing)	PATH			dimensional_characteristic_representation <= {dimensional_characteristic_representation dimensional_characteristic_representation.dimension -> dimensional_characteristic = dimensional_size} shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item => measure_representation_item

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
dimension_ tolerance to geometric_ dimension (as is_ tolerancing)	PATH			dimensional_characteristic_representation dimensional_characteristic_representation.dimension ->(dimensional_characteristic = dimensional_location dimensional_location) (dimensional_characteristic = dimensional_size dimensional_size)
dimension_ tolerance to multi_language_ string (as name)	PATH		161	shape_dimension_representation multi_language_attribute_item = shape_dimension_representation multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation.description'}
GENERAL_ TOLERANCE_TABLE	default_tolerance_ table	214		default_tolerance_table <= representation {representation.context_of_items -> representation_context representation_context.context_type = 'default setting'}
tolerance_type	representation.name	43		default_tolerance_table <= representation representation.name
GENERAL_ TOLERANCES	(default_tolerance_ table) (applied_document_ reference)	214 214		(default_tolerance_table <= representation) (applied_document_reference <= document_reference {document_reference.assigned_document -> document document.kind -> document_type document_type.product_data_type = 'general tolerances'})

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
general_ tolerances to external_ document (as table_ definition)	IDENTICAL MAPPING			
general_ tolerances to general_ tolerance_table (as table_ definition)	IDENTICAL MAPPING			
GEOMETRIC_ DIMENSION	(dimensional_ location) (dimensional_size)	47 47		
dimension_value	([measure_ representation_item] [length_measure_with_ unit]) ([measure_ representation_item] [plane_angle_measure_ with_unit]) ([measure_ representation_item] [solid_angle_measure_ with_unit])	45 41 45 41 45 41		(dimensional_location dimensional_characteristic = dimensional_location) (dimensional_size dimensional_characteristic = dimensional_size) dimensional_characteristic <- dimensional_characteristic_representation.dimension dimensional_characteristic_representation dimensional_characteristic_representation.representation -> shape_dimension_representation <= shape_representation <= representation representation.items[i] -> representation_item => [measure_representation_item] [measure_representation_item <= measure_with_unit => (length_measure_with_unit) (plane_angle_measure_with_unit) (solid_angle_measure_with_unit)]

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
id	(shape_aspect_ relationship.name) (dimensional_size. name)	41 47		(dimensional_location <= shape_aspect_relationship shape_aspect_relationship.name) (dimensional_size.name)
name	(shape_aspect_ relationship. description) (shape_aspect. description)	41 41		(dimensional_location <= shape_aspect_relationship shape_aspect_relationship.description) (dimensional_size dimensional_size.applies_to -> shape_aspect shape_aspect.description)
geometric_ dimension to multi_language_ string (as name)	PATH		102,103	(dimensional_location multi_language_attribute_item = dimensional_location multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'shape_aspect_relationship.description'}) (dimensional_size multi_language_attribute_item = dimensional_size multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'applies_to.shape_aspect.description'})
LIMITS_AND_FITS	limits_and_fits	47		
deviation	limits_and_fits.form_ variance	47		
fitting_type	limits_and_fits.zone_ variance	47		

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
grade	limits_and_fits. grade	47		
LINEAR_DISTANCE_ DIMENSION	(directed_ dimensional_ location) (dimensional_ location)	214 47		directed_dimensional_location <= dimensional_location
directed	IDENTICAL MAPPING			
orientation	placement	42		(directed_dimensional_location <= dimensional_location <=) (dimensional_location <=) shape_aspect_relationship shape_definition = shape_aspect_relationship shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement
LOCATION_ DIMENSION	dimensional_location	47		
location_ dimension to annotation_curve (as origin)	PATH			dimensional_location <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect -> shape_aspect represented_definition = shape_aspect represented_definition <-

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->representation representation.items[i] ->representation_item => styled_item => annotation_occurrence => annotation_curve_occurrence
location_dimension to constructive_geometric_element (as origin)	PATH			dimensional_location <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect ->shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->representation representation.items[i] ->representation_item => geometric_representation_item => (placement =>axis2_placement_2d) (point =>cartesian_point) (point =>point_on_curve) (curve)
location_dimension to derived_geometry (as origin)	PATH			dimensional_location <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect ->shape_aspect => derived_shape_aspect

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
location_ dimension to shape_aspect (as origin)	PATH			dimensional_location <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect -> shape_aspect
location_ dimension to annotation_curve (as target)	PATH			dimensional_location <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => styled_item => annotation_occurrence => annotation_curve_occurrence
location_ dimension to constructive_ geometric_ element (as target)	PATH			dimensional_location <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => geometric_representation_item => (placement => axis2_placement_2d) (point => cartesian_point)

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
				(point => point_on_curve) (curve)
location_ dimension to derived_geometry (as target)	PATH			dimensional_location <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect ->\ shape_aspect => derived_shape_aspect
location_ dimension to shape_aspect (as target)	PATH			dimensional_location <= shape_aspect_relationship shape_aspect_relationship.related_shape_aspect ->\ shape_aspect
PLUS_MINUS_ TOLERANCE	(tolerance_value) (qualified_ representation_item)	47 45	48	
lower_bound	(tolerance_value. lower_bound) (standard_ uncertainty. uncertainty_value)	47 45		qualified_representation_item {qualified_representation_item <= representation_item} qualified_representation_item.qualifiers[i] ->\ value_qualifier value_qualifier = uncertainty_qualifier uncertainty_qualifier.measure_name = 'lower limit' uncertainty_qualifier => standard_uncertainty standard_uncertainty.uncertainty_value
upper_bound	(tolerance_value. upper_bound) (standard_ uncertainty. uncertainty_value)	47 45		qualified_representation_item {qualified_representation_item <= representation_item} qualified_representation_item.qualifiers[i] ->\ value_qualifier value_qualifier = uncertainty_qualifier uncertainty_qualifier.measure_name = 'upper limit' uncertainty_qualifier => standard_uncertainty standard_uncertainty.uncertainty_value

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
PLUS_MINUS_ TOLERANCED_DATUM	measure_ representation_item	45		{measure_representation_item <= representation_item representation_item.name = 'plus minus tolerance value'}
tolerance_value	IDENTICAL MAPPING			
SIGNIFICANT_ NUMBER_OF_DIGITS	measure_ representation_item	45		{measure_representation_item <= representation_item representation_item.name = 'significant number of digits'}
number_of_digits	IDENTICAL MAPPING			{measure_representation_item <= measure_with_unit measure_with_unit.value_component -> measure_value measure_value = count_measure}
SIZE_DIMENSION	dimensional_size	47	26	dimensional_size
dimension_type	dimensional_size. name	47		{(dimensional_size.name = 'diameter') (dimensional_size.name = 'radius') (dimensional_size.name = 'curve length') (dimensional_size.name = 'linear')}
size_dimension to annotation_ curve (as is_applied_ to)	PATH			dimensional_size dimensional_size.applies_to -> shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => styled_item => annotation_occurrence => annotation_curve_occurrence

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
size_dimension to constructive_ geometric_ element (as is_applied_ to)	PATH			dimensional_size dimensional_size.applies_to ->\ shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->\ representation representation.items[i] ->\ representation_item =>\ geometric_representation_item =>\ (placement =>\ axis2_placement_2d) (point =>\ cartesian_point) (point =>\ point_on_curve) (curve)
size_dimension to shape_aspect (as is_applied_ to)	PATH			dimensional_location <= shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect ->\ shape_aspect
TOLERANCE_TABLE_ CELL	default_tolerance_ table_cell	214		default_tolerance_table_cell <= representation {representation.context_of_items ->\ representation_context representation_context.context_type = 'default setting'}
description	representation. description	43		default_tolerance_table_cell <= representation representation.description
tolerance_class	representation.name	43		default_tolerance_table_cell <= representation representation.name

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
tolerance_table_ cell to plus_ minus_toleranced_ datum (as cell_value)	PATH			default_tolerance_table_cell <= representation representation.items[i] -> representation_item => {representation_item.name = 'plus minus tolerance value'} measure_representation_item
tolerance_table_ cell to upper_ lower_toleranced_ datum (as cell_value)	PATH			default_tolerance_table_cell <= representation representation.items[i] -> [representation_item => {representation_item.name = 'lower tolerance value'} measure_representation_item] [representation_item => {representation_item.name = 'upper tolerance value'} measure_representation_item]
tolerance_table_ cell to multi_ language_string (as description)	PATH		100	default_tolerance_table_cell multi_language_attribute_item = default_tolerance_table_cell multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = \ 'representation.description'}
tolerance_table_ cell to significant_ number_of_digits (as limit)	PATH			default_tolerance_table_cell <= representation representation.items[i] -> representation_item => {representation_item.name = 'significant number of digits'} measure_representation_item
tolerance_table_ cell to upper_ lower_limit (as limit)	PATH			default_tolerance_table_cell <= representation representation.items[i] -> [representation_item => {representation_item.name = 'lower limit'} measure_representation_item]

Table 31 – Mapping table dimension_tolerance UoF (T1) (continued)

Application element	AIM element	Source	Rules	Reference path
				[representation_item => {representation_item.name = 'upper limit'} measure_representation_item]
tolerance_table_ cell to general_ tolerance_table (as table)	PATH			default_tolerance_table_cell <= representation <- representation_relationship.rep_2 representation_relationship representation_relationship.rep_1 -> representation => default_tolerance_table
UPPER_LOWER_ LIMIT	[measure_ representation_item] [measure_ representation_item]	45 45		{[measure_representation_item <= representation_item representation_item.name = 'lower limit'] [measure_representation_item <= representation_item representation_item.name = 'upper limit']}
lower_limit	IDENTICAL MAPPING			{measure_representation_item <= representation_item representation_item.name = 'lower limit'}
upper_limit	IDENTICAL MAPPING			{measure_representation_item <= representation_item representation_item.name = 'upper limit'}
UPPER_LOWER_ TOLERANCED_DATUM	[measure_ representation_item] [measure_ representation_item]	45 45		{[measure_representation_item <= representation_item representation_item.name = 'lower tolerance value'] [measure_representation_item <= representation_item representation_item.name = 'upper tolerance value']}
lower_tolerance_ value	IDENTICAL MAPPING			{measure_representation_item <= representation_item representation_item.name = 'lower tolerance value'}
upper_tolerance_ value	IDENTICAL MAPPING			{measure_representation_item <= representation_item representation_item.name = 'upper tolerance value'}
VALUE_LIMIT	measure_ representation_item	45		

Table 31 – Mapping table dimension_tolerance UoF (T1) (concluded)

Application element	AIM element	Source	Rules	Reference path
limit	[measure_with_unit.value_component] [measure_with_unit.unit_component]	41 41		measure_representation_item <= measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
limit_qualifier	type_qualifier.name	45		measure_representation_item <= representation_item => qualified_representation_item qualified_representation_item.qualifiers[i] -> value_qualifier = type_qualifier type_qualifier.name
VALUE_RANGE	shape_dimension_representation	47		
lower_limit	measure_representation_item	45		shape_dimension_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'lower range'} representation_item => measure_representation_item
upper_limit	measure_representation_item	45		shape_dimension_representation <= representation representation.items[i] -> {representation_item representation_item.name = 'upper range'} representation_item => measure_representation_item

Table 32 – Mapping table geometric_tolerance UoF (T2)

Application element	AIM element	Source	Rules	Reference path
ANGULARITY_ TOLERANCE	angularity_tolerance	519	61	angularity_tolerance <= geometric_tolerance_with_datum_reference
segment_size	[measure_with_unit. value_component] [measure_with_unit. unit_component]	41 41		angularity_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance => geometric_tolerance_with_defined_unit geometric_tolerance_with_defined_unit.unit_size -> measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
angularity_ tolerance to datum (as datum)	PATH			angularity_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference
CIRCULAR_RUNOUT_ TOLERANCE	circular_runout_ tolerance	519	61	circular_runout_tolerance <= geometric_tolerance_with_datum_reference
angle	[measure_with_unit. value_component] [measure_with_unit. unit_component]	41 41		circular_runout_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance <- tolerance_zone.defining_tolerance[i] tolerance_zone <- tolerance_zone_definition.zone tolerance_zone_definition => runout_zone_definition runout_zone_definition.orientation -> runout_zone_orientation runout_zone_orientation.angle -> measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
circular_runout_ tolerance to datum (as datum)	PATH			circular_runout_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
CIRCULARITY_ TOLERANCE	circularity_ tolerance	519	61	circularity_tolerance <= geometric_tolerance
COMPOUND_DATUM	datum	519		
compound_datum to datum_feature (as is_made_up_ by)	PATH			datum <= shape_aspect <- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship {shape_aspect_relationship.name = 'compound datum'} shape_aspect_relationship.related_shape_aspect -> shape_aspect => datum_feature
CONCENTRICITY_ TOLERANCE	concentricity_ tolerance	519	61	concentricity_tolerance <= geometric_tolerance_with_datum_reference
concentricity_ tolerance to datum (as datum)	PATH			concentricity_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference
CYLINDRICITY_ TOLERANCE	cylindricity_ tolerance	519	61	cylindricity_tolerance <= geometric_tolerance
DATUM	datum_reference	519		
precedence	datum_reference. precedence	519		datum_reference datum_reference.precedence
DATUM_FEATURE	(datum) ([datum] [datum_feature])	519 519 519		{(datum) (datum <= shape_aspect <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect -> shape_aspect => datum_feature)}
name	shape_aspect.name	41		datum <= shape_aspect shape_aspect.name

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
datum_feature to material_ condition (as is_ controlled_by)	PATH			datum <- datum_reference.referenced_datum datum_reference => referenced_modified_datum referenced_modified_datum.modifier -> limit_condition
datum_feature to datum_target_set (as is_defined_ by)	PATH			datum datum.established_by_relationships
datum_feature to derived_geometry (as is_defined_ by)	IDENTICAL MAPPING			
datum_feature to shape_aspect (as is_defined_ by)	IDENTICAL MAPPING			
DATUM_TARGET	datum_target	519		
id	datum_target.target_ id	519		
DATUM_TARGET_SET	datum.established_by_ relationships	519		{datum.established_by_relationships[i] -> shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect -> shape_aspect => datum_target}
rule_description	shape_aspect_ relationship. description	41		datum.established_by_relationships[i] -> shape_aspect_relationship shape_aspect_relationship.description
datum_target_set to multi_ language_string (as rule_ description)	PATH		160	shape_aspect_relationship multi_language_attribute_item = shape_aspect_relationship multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
				multi_language_attribute_assignment.attribute_name = 'description'}
datum_target_set to datum_target (as target)	PATH			datum.established_by_relationships datum <- shape_aspect_relationship.related_shape_aspect shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect ->\ shape_aspect =>\ datum_target
DERIVED_GEOMETRY	derived_shape_aspect	47		
role	shape_aspect.name	41		derived_shape_aspect <= shape_aspect shape_aspect.name
derived_geometry to constructive_ geometric_ element (as geometry)	PATH			derived_shape_aspect <= shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->\ representation representation.items[i] ->\ representation_item =>\ geometric_representation_item =>\ (placement =>\ axis2_placement_2d) (point =>\ cartesian_point) (point =>\ point_on_curve) (curve)
derived_geometry to shape_aspect (as is_derived_ from)	PATH			derived_shape_aspect <= shape_aspect <- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship {shape_aspect_relationship =>\ shape_aspect_deriving_relationship}

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship.related_shape_aspect -> shape_aspect
FLATNESS_ TOLERANCE	flatness_tolerance	519	61	flatness_tolerance <= geometric_tolerance
segment_size	[measure_with_unit. value_component] [measure_with_unit. unit_component]	41 41		flatness_tolerance <= geometric_tolerance => geometric_tolerance_with_defined_unit geometric_tolerance_with_defined_unit.unit_size -> measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
GEOMETRIC_ TOLERANCE	geometric_tolerance	519	61	{(geometric_tolerance) (<geometric_tolerance => geometric_tolerance_with_datum_reference> <geometric_tolerance => geometric_tolerance_with_defined_unit> <geometric_tolerance => modified_geometric_tolerance>)}
name	geometric_tolerance. name	519		
qualifying_note	geometric_tolerance. description	519		
significant_ digits	precision_qualifier. precision_value	45		geometric_tolerance geometric_tolerance.magnitude -> measure_with_unit <- measure_qualification.qualified_measure measure_qualification measure_qualification.qualifiers[1] -> value_qualifier value_qualifier = precision_qualifier precision_qualifier precision_qualifier.precision_value
tolerance_value	[measure_with_unit. value_component] [measure_with_unit. unit_component]	41 41		geometric_tolerance geometric_tolerance.magnitude -> measure_with_unit [measure_with_unit.value_component]

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
				[measure_with_unit.unit_component]
value_ interpretation	representation_ context.context_type	43		geometric_tolerance geometric_tolerance.magnitude -> measure_with_unit => measure_representation_item <= representation_item <- representation representation.context_of_items -> representation_context representation_context.context_type {(representation_context.context_type) (representation_context.context_type = 'required') (representation_context.context_type= 'calculated') (representation_context.context_type= 'measured') (representation_context.context_type= 'estimated')}
geometric_ tolerance to annotation_curve (as is_applied_ to)	PATH			geometric_tolerance geometric_tolerance.toleranced_shape_aspect -> shape_aspect represented_definition = shape_aspect represented_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> representation_item => styled_item => annotation_occurrence => annotation_curve_occurrence

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_ tolerance to constructive_ geometric_ element (as is_applied_ to)	PATH			geometric_tolerance geometric_tolerance.toleranced_shape_aspect ->shape_aspect represented_definition = shape_aspect represented_definition <-property_definition_representation.definition property_definition_representation property_definition_representation.used_representation ->representation representation.items[i] ->representation_item => geometric_representation_item => (placement =>axis2_placement_2d) (point =>cartesian_point) (point =>point_on_curve) (curve)
geometric_ tolerance to derived_geometry (as is_applied_ to)	PATH			geometric_tolerance geometric_tolerance.toleranced_shape_aspect ->derived_shape_aspect
geometric_ tolerance to shape_aspect (as is_applied_ to)	PATH			geometric_tolerance geometric_tolerance.toleranced_shape_aspect ->shape_aspect
geometric_ tolerance to material_ condition (as is_ controlled_by)	PATH			geometric_tolerance =>modified_geometric_tolerance modified_geometric_tolerance.modifier ->limit_condition

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
geometric_ tolerance to multi_language_ string (as name)	PATH		115	geometric_tolerance multi_language_attribute_item = geometric_tolerance multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'name'}
geometric_ tolerance to multi_language_ string (as qualifying_ note)	PATH		115	geometric_tolerance multi_language_attribute_item = geometric_tolerance multi_language_attribute_item <- applied_multi_language_attribute_assignment.items[1] applied_multi_language_attribute_assignment {applied_multi_language_attribute_assignment <= multi_language_attribute_assignment multi_language_attribute_assignment.attribute_name = 'description'}
GEOMETRIC_ TOLERANCE_ PRECEDENCE	geometric_tolerance_ relationship	519		{geometric_tolerance_relationship.name = 'precedence'}
geometric_ tolerance_ precedence to geometric_ tolerance (as primary_ tolerance)	PATH			geometric_tolerance_relationship geometric_tolerance_relationship.relating_geometric_tolerance -> geometric_tolerance
geometric_ tolerance_ precedence to geometric_ tolerance (as secondary_ tolerance)	PATH			geometric_tolerance_relationship geometric_tolerance_relationship.related_geometric_tolerance -> geometric_tolerance
LINEAR_PROFILE_ TOLERANCE	linear_profile_ tolerance	519	61	linear_profile_tolerance <= geometric_tolerance

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
linear_profile_ tolerance to reference_plane (as affected_ plane)	PATH			linear_profile_tolerance <= geometric_tolerance <- tolerance_zone.defining_tolerance[i] tolerance_zone <= shape_aspect <- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'affected plane'} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'affected plane'} representation_item => geometric_representation_item => placement
linear_profile_ tolerance to datum (as datum)	PATH			linear_profile_tolerance <= geometric_tolerance => geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference
MATERIAL_ CONDITION	limit_condition	47		
condition	IDENTICAL MAPPING			

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
PARALLELISM_ TOLERANCE	parallelism_ tolerance	519	61	parallelism_tolerance <= geometric_tolerance_with_datum_reference
segment_size	[measure_with_unit. value_component] [measure_with_unit. unit_component]	41 41		parallelism_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance => geometric_tolerance_with_defined_unit geometric_tolerance_with_defined_unit.unit_size -> measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
parallelism_ tolerance to reference_plane (as affected_ plane)	PATH			parallelism_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance <- tolerance_zone.defining_tolerance[i] tolerance_zone <= shape_aspect <- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'affected plane'} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'affected plane'} representation_item =>

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
				geometric_representation_item => placement
parallelism_ tolerance to datum (as datum)	PATH			parallelism_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference
PERPENDICULARITY_ TOLERANCE	perpendicularity_ tolerance	519	61	perpendicularity_tolerance <= geometric_tolerance_with_datum_reference
segment_size	[measure_with_unit. value_component] [measure_with_unit. unit_component]	41 41		perpendicularity_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance => geometric_tolerance_with_defined_unit geometric_tolerance_with_defined_unit.unit_size -> measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
perpendicularity_ tolerance to reference_plane (as affected_ plane)	PATH			perpendicularity_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance <- tolerance_zone.defining_tolerance[i] tolerance_zone <= shape_aspect <- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'affected plane'} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
				property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'affected plane'} representation_item => geometric_representation_item => placement
perpendicularity_tolerance to datum (as datum)	PATH			perpendicularity_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference
PLACED_TARGET	placed_datum_target_feature	214		placed_datum_target_feature <= datum_target {datum_target <= shape_aspect (shape_aspect.description = 'point') (shape_aspect.description = 'line') (shape_aspect.description = 'rectangle') (shape_aspect.description = 'circle')}
placed_target to cartesian_coordinate_space (as is_defined_in)	PATH			placed_datum_target_feature <= datum_target <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation =>

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation_with_parameters} representation representation.context_of_items ->\ representation_context => geometric_representation_context
placed_target to axis_placement (as parameter_ reference)	PATH			placed_datum_target_feature <= datum_target <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation ->\ {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] ->\ {representation_item representation_item.name = 'orientation'} representation_item => geometric_representation_item => placement
POSITION_ TOLERANCE	position_tolerance	519	61	position_tolerance <= geometric_tolerance_with_datum_reference

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
position_ tolerance to reference_plane (as affected_ plane)	PATH			position_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance <- tolerance_zone.defining_tolerance[i] tolerance_zone <= shape_aspect <- shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'affected plane'} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'affected plane'} representation_item => geometric_representation_item => placement
position_ tolerance to datum (as datum)	PATH			position_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference
PROJECTION	projected_zone_ definition	519		

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
projection_ length	[measure_with_unit. value_component] [measure_with_unit. unit_component]	41 41		projected_zone_definition projected_zone_definition.projected_length -> measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
projection to tolerance_zone (as is_ specifying)	PATH			projected_zone_definition <= tolerance_zone_definition tolerance_zone_definition.zone tolerance_zone
projection to shape_aspect (as projection_ end)	PATH			projected_zone_definition projected_zone_definition.projection_end -> shape_aspect
REFERENCE_PLANE	placement	42		
plane_location	placement.location	42		
plane_normal	(axis1_placement. axis) ((axis2_placement_3d. axis) (axis2_placement_3d. ref_direction))	42 42 42		placement => (axis1_placement axis1_placement.axis) (axis2_placement_3d (axis2_placement_3d.axis) (axis2_placement_3d.ref_direction))
STRAIGHTNESS_ TOLERANCE	straightness_ tolerance	519	61	straightness_tolerance <= geometric_tolerance
segment_size	[measure_with_unit. value_component] [measure_with_unit. unit_component]	41 41		straightness_tolerance <= geometric_tolerance => geometric_tolerance_with_defined_unit geometric_tolerance_with_defined_unit.unit_size -> measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
straightness_ tolerance to reference_plane (as affected_ plane)	PATH			straightness_tolerance <= geometric_tolerance <- tolerance_zone.defining_tolerance[i] tolerance_zone <= shape_aspect <- shape_aspect_relationship.relating_shape_aspect

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'affected plane'} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'affected plane'} representation_item => geometric_representation_item => placement
SURFACE_PROFILE_ TOLERANCE	surface_profile_ tolerance	519	61	surface_profile_tolerance <= geometric_tolerance
surface_profile_ tolerance to datum (as datum)	PATH			surface_profile_tolerance <= geometric_tolerance => geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference
SYMMETRY_ TOLERANCE	symmetry_tolerance	519	61	symmetry_tolerance <= geometric_tolerance_with_datum_reference
symmetry_ tolerance to reference_plane (as affected_ plane)	PATH			symmetry_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance <- tolerance_zone.defining_tolerance[i] tolerance_zone <= shape_aspect <-

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_aspect_relationship.relating_shape_aspect shape_aspect_relationship shape_aspect_relationship.related_shape_aspect -> shape_aspect {shape_aspect shape_aspect.description = 'affected plane'} shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition property_definition_representation property_definition_representation.used_representation -> representation representation.items[i] -> {representation_item representation_item.name = 'affected plane'} representation_item => geometric_representation_item => placement
symmetry_ tolerance to datum (as datum)	PATH			symmetry_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference
TARGET_AREA	datum_target	519		
target_area to shape_aspect (as is_defined_ by)	IDENTICAL MAPPING			
TARGET_CIRCLE	placed_datum_target_ feature	214		placed_datum_target_feature <= datum_target {datum_target <= shape_aspect shape_aspect.description = 'circle'}

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
diameter	measure_representation_item	45		placed_datum_target_feature <= datum_target <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'target diameter'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
TARGET_LINE	placed_datum_target_feature	214		placed_datum_target_feature <= datum_target {datum_target <= shape_aspect shape_aspect.description = 'line'}
target_length	measure_representation_item	45		placed_datum_target_feature <= datum_target <= shape_aspect shape_definition = shape_aspect shape_definition

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'target length'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
TARGET_POINT	placed_datum_target_feature	214		placed_datum_target_feature <= datum_target {datum_target <= shape_aspect shape_aspect.description = 'point'}
TARGET_RECTANGLE	placed_datum_target_feature	214		placed_datum_target_feature <= datum_target {datum_target <= shape_aspect shape_aspect.description = 'rectangle'}
target_length	measure_representation_item	45		placed_datum_target_feature <= datum_target <= shape_aspect shape_definition = shape_aspect shape_definition

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
				characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation => shape_representation => shape_representation_with_parameters} representation representation.items[i] -> {representation_item representation_item.name = 'target length'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
target_width	measure_ representation_item	45		placed_datum_target_feature <= datum_target <= shape_aspect shape_definition = shape_aspect shape_definition characterized_definition = shape_definition characterized_definition <- property_definition.definition property_definition <- property_definition_representation.definition {property_definition_representation => shape_definition_representation} property_definition_representation property_definition_representation.used_representation -> {representation =>

Table 32 – Mapping table geometric_tolerance UoF (T2) (continued)

Application element	AIM element	Source	Rules	Reference path
				shape_representation => shape_representation_with_parameters} representation representation.items[i] ->{representation_item representation_item.name = 'target width'} representation_item => measure_representation_item {measure_representation_item <= measure_with_unit => length_measure_with_unit}
TOLERANCE_ZONE	tolerance_zone	519		
form_type	tolerance_zone_form.name	519		tolerance_zone tolerance_zone.form ->tolerance_zone_form tolerance_zone_form.name
tolerance_zone to geometric_tolerance (as is_zone_for)	PATH			tolerance_zone tolerance_zone.defining_tolerance[i] ->geometric_tolerance
TOLERANCE_ZONE_DEFINITION	tolerance_zone_definition	47		{(tolerance_zone_definition) (tolerance_zone_definition =>projected_zone_definition) (tolerance_zone_definition =>runout_zone_definition)}
tolerance_zone_definition to shape_aspect (as first_element)	PATH			tolerance_zone_definition tolerance_zone_definition.boundaries[1] ->shape_aspect
tolerance_zone_definition to tolerance_zone (as is_defining)	PATH			tolerance_zone_definition tolerance_zone_definition.zone ->tolerance_zone

Table 32 – Mapping table geometric_tolerance UoF (T2) (concluded)

Application element	AIM element	Source	Rules	Reference path
tolerance_zone_ definition to shape_aspect (as second_ element)	PATH			tolerance_zone_definition tolerance_zone_definition.boundaries[2] -> shape_aspect
TOTAL_RUNOUT_ TOLERANCE	total_runout_ tolerance	519	61	total_runout_tolerance <= geometric_tolerance_with_datum_reference
angle	[measure_with_unit. value_component] [measure_with_unit. unit_component]	41 41		total_runout_tolerance <= geometric_tolerance_with_datum_reference <= geometric_tolerance <- tolerance_zone.defining_tolerance[i] tolerance_zone <- tolerance_zone_definition.zone tolerance_zone_definition => runout_zone_definition runout_zone_definition.orientation -> runout_zone_orientation runout_zone_orientation.angle -> measure_with_unit [measure_with_unit.value_component] [measure_with_unit.unit_component]
total_runout_ tolerance to datum (as datum)	PATH			total_runout_tolerance <= geometric_tolerance_with_datum_reference geometric_tolerance_with_datum_reference.datum_system[i] -> datum_reference

The following rules are referenced in the preceding table:

1 action_assignment_subtype_exclusiveness

2 action_subtype_restriction

3 activity_requires_person_or_organization_assignment

4 application_context_requires_ap_definition

5 approval_person_organization_requires_date_time

6 approval_requires_approval_assignment

7 characterized_object_subtype_exclusiveness

8 configuration_design_subtype_exclusiveness

9 coordinated_assembly_and_shape

10 curve_font_usage

11 definition_replacement_requires_effectivity_assignment

12 dependent_instantiable_action_request_role

13 dependent_instantiable_action_role

14 dependent_instantiable_approval_date_time_role

15 dependent_instantiable_approval_role

16 dependent_instantiable_approver_role

17 dependent_instantiable_certification_role

18 dependent_instantiable_colour_rgb

19 dependent_instantiable_contract_role

20 dependent_instantiable_curve_style

21 dependent_instantiable_date

22 dependent_instantiable_date_and_time

23 dependent_instantiable_date_role

24 dependent_instantiable_date_time_role

25 dependent_instantiable_derived_unit

26 dependent_instantiable_dimensional_size

27 dependent_instantiable_document_role

28 dependent_instantiable_effectivity_role

29 dependent_instantiable_externally_defined_symbol

30 dependent_instantiable_fill_area_style

31 dependent_instantiable_fill_area_style_colour

32 dependent_instantiable_group_role

33 dependent_instantiable_identification_role

34 dependent_instantiable_language_role

35 dependent_instantiable_measure_with_unit

36 dependent_instantiable_named_unit

37 dependent_instantiable_organization_role

38 dependent_instantiable_person_and_organization_role

39 dependent_instantiable_person_role

40 dependent_instantiable_pre_defined_colour

41 dependent_instantiable_pre_defined_symbol

42 dependent_instantiable_presentation_style_by_context

43 dependent_instantiable_representation

44 dependent_instantiable_security_classification_role

45 dependent_instantiable_symbol_colour

46 dependent_instantiable_text_style

47 dependent_instantiable_text_style_for_defined_font

48 dependent_instantiable_tolerance_value

49 dependent_instantiable_uncertainty_qualifier

50 dimensionality_is_two_or_three

51 draughting_model_annotation_layers

52 draughting_subfigure_representation_layers

53 drawing_sheet_annotation_layers

54 drawing_sheet_layout_usage

55 drawing_view_annotation_layers

56 effective_context_assignment_subtype_exclusiveness

57 effective_item_assignment_subtype_exclusiveness

58 effectivity_requires_effective_item_assignment

59 fill_area_style_tile_symbol_constraint

60 functional_product_definition_requires_functional_product

61 geometric_tolerance_subtype_exclusiveness

62 global_length_and_angle_units_2d_or_3d

63 group_assignment_subtype_exclusiveness

64 identification_assignment_subtype_exclusiveness

65 organization_assignment_subtype_exclusiveness

66 person_requires_person_and_organization

67 presentation_layer_assignment_constraint_2d_or_3d

68 product_definition_substitute_requires_effectivity_assignment

69 product_definition_subtype_exclusiveness

70 product_requires_version

71 representation_subtype_exclusiveness

72 restrict_action_directive

73 restrict_action_resource_requirement_for_process_operation

74 restrict_approval_date_time_for_approval

75 restrict_approval_date_time_role

76 restrict_classification_system_for_approval_status

77 restrict_classification_system_for_document_type

78 restrict_classification_system_for_security_classification_level

79 restrict_multi_language_for_action

80 restrict_multi_language_for_action_directive

81 restrict_multi_language_for_action_method

82 restrict_multi_language_for_action_property

83 restrict_multi_language_for_action_relationship

84 restrict_multi_language_for_action_resource_requirement

85 restrict_multi_language_for_alternate_product_relationship

86 restrict_multi_language_for_application_context

87 restrict_multi_language_for_approval

88 restrict_multi_language_for_assembly_component_usage

89 restrict_multi_language_for_assembly_component_usage_substitute

90 restrict_multi_language_for_camera_image_2d_with_scale

91 restrict_multi_language_for_camera_image_3d_with_scale

92 restrict_multi_language_for_certification

93 restrict_multi_language_for_characterized_product_category

94 restrict_multi_language_for_classification_system

95 restrict_multi_language_for_configurable_item

96 restrict_multi_language_for_configuration_design

97 restrict_multi_language_for_configuration_item_relationship

98 restrict_multi_language_for_constructive_geometry_representation_relationship

99 restrict_multi_language_for_data_environment

100 restrict_multi_language_for_default_tolerance_table_cell

101 restrict_multi_language_for_descriptive_representation_item

102 restrict_multi_language_for_dimensional_location

103 restrict_multi_language_for_dimensional_size

104 restrict_multi_language_for_document

105 restrict_multi_language_for_document_relationship

106 restrict_multi_language_for_draughting_model

107 restrict_multi_language_for_draughting_title

108 restrict_multi_language_for_drawing_sheet_revision

109 restrict_multi_language_for_effectivity

110 restrict_multi_language_for_effectivity_relationship

111 restrict_multi_language_for_externally_defined_classification_attribute

112 restrict_multi_language_for_externally_defined_classification_system

113 restrict_multi_language_for_externally_defined_product_category

114 restrict_multi_language_for_externally_defined_representation

115 restrict_multi_language_for_geometric_tolerance

116 restrict_multi_language_for_group

117 restrict_multi_language_for_identification_role

118 restrict_multi_language_for_make_from_usage_option

119 restrict_multi_language_for_material_property

120 restrict_multi_language_for_measure_representation_item

121 restrict_multi_language_for_mechanical_design_geometric_presentation_representation

122 restrict_multi_language_for_model_change

123 restrict_multi_language_for_organization_relationship

124 restrict_multi_language_for_organization_role

125 restrict_multi_language_for_pair_actuator

126 restrict_multi_language_for_presentation_area

127 restrict_multi_language_for_presentation_layer_assignment

128 restrict_multi_language_for_presentation_view

129 restrict_multi_language_for_presentation_view_relationship

130 restrict_multi_language_for_process_change

131 restrict_multi_language_for_process_plan

132 restrict_multi_language_for_process_product_association

133 restrict_multi_language_for_product

134 restrict_multi_language_for_product_category

135 restrict_multi_language_for_product_class

136 restrict_multi_language_for_product_component

137 restrict_multi_language_for_product_concept_feature

138 restrict_multi_language_for_product_concept_feature_association

139 restrict_multi_language_for_product_concept_feature_category

140 restrict_multi_language_for_product_definition

141 restrict_multi_language_for_product_definition_formation

142 restrict_multi_language_for_product_definition_formation_relationship

143 restrict_multi_language_for_product_definition_occurrence

144 restrict_multi_language_for_product_definition_relationship

145 restrict_multi_language_for_product_definition_shape

146 restrict_multi_language_for_product_definition_substitute

147 restrict_multi_language_for_project

148 restrict_multi_language_for_property_change

149 restrict_multi_language_for_property_definition

150 restrict_multi_language_for_property_definition_relationship

151 restrict_multi_language_for_property_definition_representation

152 restrict_multi_language_for_rep_item_group

153 restrict_multi_language_for_representation

154 restrict_multi_language_for_representation_item

155 restrict_multi_language_for_representation_relationship

156 restrict_multi_language_for_resource_requirement_type

157 restrict_multi_language_for_retention

158 restrict_multi_language_for_security_classification

159 restrict_multi_language_for_shape_aspect

160 restrict_multi_language_for_shape_aspect_relationship

161 restrict_multi_language_for_shape_dimension_representation

162 restrict_multi_language_for_shape_representation

163 restrict_multi_language_for_solution_based_product_definition

164 restrict_multi_language_for_versioned_action_request

165 restrict_process_product_association

166 restrict_product_categories_for_tool_part_relationship

167 restrict_product_concept_features_for_product_specification

168 restrict_version_assignment_for_action_directive

169 restrict_version_assignment_for_configurable_item

170 restrict_version_assignment_for_document

171 restrict_version_assignment_for_draughting_model

172 restrict_version_assignment_for_mechanical_design_geometric_presentation_representation

173 restrict_version_assignment_for_physically_modelled_product_definition

174 restrict_version_assignment_for_presentation_area

175 restrict_version_assignment_for_process_plan

176 restrict_version_assignment_for_product_class

177 restrict_version_assignment_for_product_component

178 restrict_version_assignment_for_product_concept_feature

179 restrict_version_assignment_for_product_definition

180 restrict_version_assignment_for_representation

181 restrict_version_assignment_for_shape_representation

182 restrict_version_assignment_for_solution_based_product_definition

183 retention_requires_retention_assignment

184 security_classification_requires_security_classification_assignment

185 shape_aspect_relationship_subtype_exclusiveness

186 shape_aspect_subtype_exclusiveness

187 shape_representation_subtype_exclusiveness

188 sheets_belong_to_one_drawing

189 styled_curve

190 subtype_mandatory_camera_image_2d_or_3d

191 subtype_mandatory_camera_model_2d_or_3d

192 subtype_mandatory_drawing_revision

193 subtype_mandatory_effectivity

194 subtype_mandatory_pre_defined_colour

195 subtype_mandatory_pre_defined_curve_font

196 subtype_mandatory_pre_defined_text_font

197 subtype_mandatory_product_concept

198 subtype_mandatory_product_definition_with_associated_documents

199 terminator_symbol_constraint

200 text_font_usage

201 versioned_action_request_requires_date_and_person_or_organization

5.2 AIM EXPRESS short listing

This clause specifies the EXPRESS schema that uses elements from the integrated resources (and the AICs) and contains the types, entity specializations, rules, and functions that are specific to this part of ISO 10303. This clause also specifies modifications to the text for constructs that are imported from the integrated resources (and the AICs). The definitions and EXPRESS provided in the integrated resources for constructs used in the AIM may include select list items and subtypes that are not imported into the AIM. Requirements stated in the integrated resources that refer to such items and subtypes apply exclusively to those items that are imported into the AIM.

NOTE 1 – The following conventions apply for the definition text:

– Terms written in boldface refer to the AIM objects, i.e., '**circle**' refers to the entity as defined in ISO 10303-42;

– Terms written in normal face refer to the concept, i.e., 'circle' refers to the concept of a round plane figure whose circumference is everywhere equidistant from its centre.

NOTE 2 – The following shortform and the longform documented in annex A correspond to the conformance class CC8 of this part of ISO 10303 which includes all UoFs. This document does not contain shortforms or longforms for the other conformance classes although this is planned for the future.

EXPRESS specification:

```
*)
SCHEMA automotive_design;
```

```
USE FROM application_context_schema   -- ISO 10303-41
 (application_context,
  application_context_element,
  application_context_element_relationship,
  application_protocol_definition,
  configuration_item_context,
  product_definition_context);

USE FROM product_definition_schema   -- ISO 10303-41
 (product,
  product_category,
  product_category_relationship,
  product_definition,
  product_definition_formation,
  product_definition_formation_relationship,
  product_definition_relationship,
  product_definition_substitute,
  product_definition_with_associated_documents,
  product_related_product_category);

USE FROM product_property_definition_schema   -- ISO 10303-41
 (characterized_object,
  characterized_object_relationship,
  product_definition_shape,
  property_definition,
  shape_aspect,
  shape_aspect_relationship);

USE FROM product_property_representation_schema   -- ISO 10303-41
 (context_dependent_shape_representation,
  property_definition_representation,
  shape_definition_representation,
  shape_representation,
  shape_representation_relationship);

USE FROM management_resources_schema   -- ISO 10303-41
 (action_assignment,
  action_request_assignment,
  action_request_role,
  action_role,
  approval_assignment,
  approval_role,
  certification_assignment,
  certification_role,
  contract_assignment,
  contract_role,
  date_and_time_assignment,
  date_assignment,
  date_role,
  date_time_role,
```

```
  document_reference,
  document_role,
  effective_context_assignment,
  effective_item_assignment,
  effectivity_assignment,
  effectivity_role,
  external_identification_assignment,
  group_assignment,
  group_role,
  identification_assignment,
  identification_role,
  language_role,
  multi_language_attribute_assignment,
  organization_assignment,
  organization_role,
  person_and_organization_assignment,
  person_and_organization_role,
  person_assignment,
  person_role,
  security_classification_assignment,
  security_classification_role);

USE FROM document_schema   -- ISO 10303-41
 (document,
  document_relationship,
  document_type,
  document_usage_constraint);

USE FROM action_schema   -- ISO 10303-41
 (action,
  action_directive,
  action_method,
  action_relationship,
  action_request_solution,
  action_request_status,
  action_resource,
  action_status,
  directed_action,
  executed_action,
  versioned_action_request);

USE FROM certification_schema   -- ISO 10303-41
 (certification,
  certification_type);

USE FROM approval_schema   -- ISO 10303-41
 (approval,
  approval_date_time,
  approval_date_time_role,
  approval_person_organization,
  approval_relationship,
  approval_status,
```

```
  approver_role);

USE FROM security_classification_schema   -- ISO 10303-41
 (security_classification,
  security_classification_level);

USE FROM contract_schema   -- ISO 10303-41
 (contract);

USE FROM person_organization_schema   -- ISO 10303-41
 (organization,
  organization_relationship,
  organizational_address,
  person,
  person_and_organization,
  personal_address);

USE FROM date_time_schema   -- ISO 10303-41
 (calendar_date,
  date,
  date_and_time,
  local_time);

USE FROM group_schema   -- ISO 10303-41
 (group,
  group_relationship);

USE FROM effectivity_schema   -- ISO 10303-41
 (dated_effectivity,
  effectivity,
  effectivity_relationship,
  lot_effectivity,
  serial_numbered_effectivity);

USE FROM external_reference_schema   -- ISO 10303-41
 (external_source,
  externally_defined_item,
  pre_defined_item);

USE FROM support_resource_schema   -- ISO 10303-41
 (identifier);

USE FROM measure_schema   -- ISO 10303-41
 (amount_of_substance_measure,
  amount_of_substance_measure_with_unit,
  amount_of_substance_unit,
  area_measure,
  area_measure_with_unit,
  area_unit,
  context_dependent_measure,
  context_dependent_unit,
  conversion_based_unit,
```

```
  count_measure,
  derived_unit,
  descriptive_measure,
  electric_current_measure,
  electric_current_measure_with_unit,
  electric_current_unit,
  expression_conversion_based_unit,
  global_unit_assigned_context,
  length_measure,
  length_measure_with_unit,
  length_unit,
  luminous_intensity_measure,
  luminous_intensity_measure_with_unit,
  luminous_intensity_unit,
  mass_measure,
  mass_measure_with_unit,
  mass_unit,
  measure_with_unit,
  named_unit,
  numeric_measure,
  parameter_value,
  plane_angle_measure,
  plane_angle_measure_with_unit,
  plane_angle_unit,
  positive_length_measure,
  positive_plane_angle_measure,
  positive_ratio_measure,
  ratio_measure,
  ratio_measure_with_unit,
  ratio_unit,
  si_unit,
  solid_angle_measure,
  solid_angle_measure_with_unit,
  solid_angle_unit,
  thermodynamic_temperature_measure,
  thermodynamic_temperature_measure_with_unit,
  thermodynamic_temperature_unit,
  time_measure,
  time_measure_with_unit,
  time_unit,
  volume_measure,
  volume_measure_with_unit,
  volume_unit);

USE FROM geometry_schema   -- ISO 10303-42
 (axis1_placement,
  axis2_placement_2d,
  axis2_placement_3d,
  cartesian_point,
  cartesian_transformation_operator,
  cartesian_transformation_operator_2d,
  cartesian_transformation_operator_3d,
```

```
  curve,
  degenerate_pcurve,
  direction,
  geometric_representation_context,
  geometric_representation_item,
  pcurve,
  placement,
  point,
  point_on_curve,
  point_on_surface,
  point_replica,
  surface,
  surface_curve);

USE FROM topology_schema   -- ISO 10303-42
 (closed_shell,
  edge,
  edge_curve,
  face,
  face_bound,
  face_outer_bound,
  face_surface,
  open_shell,
  oriented_edge,
  oriented_face,
  topological_representation_item,
  vertex,
  vertex_point);

USE FROM representation_schema   -- ISO 10303-43
 (compound_representation_item,
  functionally_defined_transformation,
  global_uncertainty_assigned_context,
  item_defined_transformation,
  list_representation_item,
  mapped_item,
  representation,
  representation_context,
  representation_item,
  representation_map,
  representation_relationship,
  representation_relationship_with_transformation,
  uncertainty_assigned_representation,
  uncertainty_measure_with_unit);

USE FROM product_structure_schema   -- ISO 10303-44
 (alternate_product_relationship,
  assembly_component_usage,
  assembly_component_usage_substitute,
  make_from_usage_option,
  next_assembly_usage_occurrence,
  product_definition_usage,
```

```
  promissory_usage_occurrence,
  quantified_assembly_component_usage,
  specified_higher_usage_occurrence);

USE FROM product_concept_schema   -- ISO 10303-44
 (concept_feature_relationship_with_condition,
  conditional_concept_feature,
  product_concept,
  product_concept_feature,
  product_concept_feature_association,
  product_concept_relationship);

USE FROM configuration_management_schema   -- ISO 10303-44
 (configurable_item,
  configuration_design,
  configuration_item,
  configuration_item_relationship,
  configuration_item_with_context);

USE FROM material_property_definition_schema   -- ISO 10303-45
 (material_designation,
  material_designation_characterization,
  material_property,
  property_definition_relationship);

USE FROM material_property_representation_schema   -- ISO 10303-45
 (data_environment,
  material_property_representation);

USE FROM qualified_measure_schema   -- ISO 10303-45
 (descriptive_representation_item,
  measure_qualification,
  measure_representation_item,
  precision_qualifier,
  qualified_representation_item,
  qualitative_uncertainty,
  standard_uncertainty,
  type_qualifier,
  uncertainty_qualifier);

USE FROM presentation_organization_schema   -- ISO 10303-46
 (area_in_set,
  camera_image,
  camera_model,
  camera_model_d2,
  camera_model_d3_with_hlhsr,
  camera_usage,
  light_source,
  light_source_ambient,
  light_source_directional,
  light_source_positional,
  light_source_spot,
```

```
  presentation_area,
  presentation_layer_assignment,
  presentation_layer_usage,
  presentation_representation,
  presentation_set,
  presentation_size,
  presentation_view,
  presented_item,
  presented_item_representation);

USE FROM presentation_definition_schema   -- ISO 10303-46
 (annotation_curve_occurrence,
  annotation_occurrence,
  pre_defined_symbol,
  text_string_representation);

USE FROM presentation_appearance_schema   -- ISO 10303-46
 (context_dependent_over_riding_styled_item,
  curve_style,
  curve_style_rendering,
  externally_defined_curve_font,
  invisibility,
  pre_defined_marker,
  styled_item,
  surface_side_style,
  surface_style_fill_area,
  surface_style_reflectance_ambient,
  surface_style_reflectance_ambient_diffuse,
  surface_style_reflectance_ambient_diffuse_specular,
  surface_style_rendering,
  surface_style_rendering_with_properties,
  surface_style_transparent,
  surface_style_usage,
  u_direction_count,
  v_direction_count);

USE FROM presentation_resource_schema   -- ISO 10303-46
 (colour,
  colour_rgb,
  planar_box,
  pre_defined_text_font);

USE FROM shape_aspect_definition_schema   -- ISO 10303-47
 (composite_shape_aspect,
  datum_target,
  derived_shape_aspect,
  extension,
  limit_condition,
  perpendicular_to,
  shape_aspect_deriving_relationship);

USE FROM shape_dimension_schema   -- ISO 10303-47
```

```
 (angular_location,
  angular_size,
  dimensional_characteristic_representation,
  dimensional_location,
  dimensional_location_with_path,
  dimensional_size,
  shape_dimension_representation);

USE FROM shape_tolerance_schema   -- ISO 10303-47
 (limits_and_fits,
  plus_minus_tolerance,
  projected_zone_definition,
  runout_zone_definition,
  tolerance_value,
  tolerance_zone_definition);

USE FROM method_definition_schema   -- ISO 10303-49
 (relationship_condition);

USE FROM process_property_schema   -- ISO 10303-49
 (action_property,
  action_resource_requirement,
  process_product_association,
  process_property_association,
  product_definition_process,
  property_process,
  requirement_for_action_resource,
  resource_property,
  resource_requirement_type);

USE FROM process_property_representation_schema   -- ISO 10303-49
 (action_property_representation,
  resource_property_representation);

USE FROM drawing_definition_schema   -- ISO 10303-101
 (draughting_title,
  drawing_definition,
  drawing_revision,
  drawing_revision_sequence,
  drawing_sheet_revision,
  drawing_sheet_revision_usage);

USE FROM draughting_element_schema   -- ISO 10303-101
 (draughting_callout);

USE FROM kinematic_structure_schema   -- ISO 10303-105
 (cylindrical_pair,
  cylindrical_pair_range,
  cylindrical_pair_value,
  fully_constrained_pair,
  gear_pair,
  gear_pair_range,
```

```
  gear_pair_value,
  initial_state,
  kinematic_ground_representation,
  kinematic_joint,
  kinematic_link,
  kinematic_link_representation,
  kinematic_link_representation_association,
  kinematic_link_representation_relation,
  kinematic_pair,
  kinematic_property_definition,
  kinematic_structure,
  mechanism,
  mechanism_base_placement,
  pair_actuator,
  pair_value,
  planar_curve_pair,
  planar_curve_pair_range,
  planar_pair_range,
  planar_pair_value,
  point_on_planar_curve_pair_range,
  point_on_planar_curve_pair_value,
  point_on_surface_pair_range,
  point_on_surface_pair_value,
  prismatic_pair_range,
  prismatic_pair_value,
  rack_and_pinion_pair_range,
  rack_and_pinion_pair_value,
  revolute_pair_range,
  revolute_pair_value,
  rolling_curve_pair,
  rolling_curve_pair_value,
  rolling_surface_pair,
  rolling_surface_pair_value,
  screw_pair_range,
  screw_pair_value,
  simple_pair_range,
  sliding_curve_pair,
  sliding_curve_pair_value,
  sliding_surface_pair,
  sliding_surface_pair_value,
  spherical_pair_range,
  spherical_pair_value,
  surface_pair_range,
  unconstrained_pair_value,
  universal_pair_range,
  universal_pair_value,
  unlimited_range);

USE FROM kinematic_analysis_control_and_result_schema   -- ISO 10303-105
 (configuration_definition,
  configuration_interpolation,
  interpolated_configuration_sequence,
```

```
  kinematic_analysis_result);


USE FROM aic_advanced_brep;   -- ISO 10303-514

USE FROM aic_associative_draughting_elements;   -- ISO 10303-520

USE FROM aic_csg;   -- ISO 10303-515

USE FROM aic_draughting_annotation;   -- ISO 10303-504

USE FROM aic_draughting_elements;   -- ISO 10303-506

USE FROM aic_edge_based_wireframe;   -- ISO 10303-501

USE FROM aic_faceted_brep;   -- ISO 10303-512

USE FROM aic_geometrical_tolerances;   -- ISO 10303-519

USE FROM aic_geometrically_bounded_2d_wireframe;   -- ISO 10303-503

USE FROM aic_geometrically_bounded_surface;   -- ISO 10303-507

USE FROM aic_geometrically_bounded_wireframe;   -- ISO 10303-510

USE FROM aic_manifold_surface;   -- ISO 10303-509

USE FROM aic_mechanical_design_geometric_presentation;   -- ISO 10303-517

USE FROM aic_non_manifold_surface;   -- ISO 10303-508

USE FROM aic_topology_bounded_surface;   -- ISO 10303-511


(*
```

NOTE 3 – The schemas referenced above can be found in the following Parts of ISO 10303:

action_schema	ISO 10303-41
aic_advanced_brep	ISO 10303-514
aic_associative_draughting_elements	ISO 10303-520
aic_csg	ISO 10303-515
aic_draughting_annotation	ISO 10303-504
aic_draughting_elements	ISO 10303-506
aic_edge_based_wireframe	ISO 10303-501
aic_faceted_brep	ISO 10303-512
aic_geometrical_tolerances	ISO 10303-519
aic_geometrically_bounded_2d_wireframe	ISO 10303-503
aic_geometrically_bounded_surface	ISO 10303-507
aic_geometrically_bounded_wireframe	ISO 10303-510
aic_manifold_surface	ISO 10303-509
aic_mechanical_design_geometric_presentation	ISO 10303-517

aic_non_manifold_surface	ISO 10303-508
aic_shell_based_wireframe	ISO 10303-502
application_context_schema	ISO 10303-41
approval_schema	ISO 10303-41
certification_schema	ISO 10303-41
configuration_management_schema	ISO 10303-44
contract_schema	ISO 10303-41
date_time_schema	ISO 10303-41
document_schema	ISO 10303-41
draughting_element_schema	ISO 10303-101
drawing_definition_schema	ISO 10303-101
effectivity_schema	ISO 10303-41
external_reference_schema	ISO 10303-41
geometry_schema	ISO 10303-42
group_schema	ISO 10303-41
kinematic_analysis_control_and_result_schema	ISO 10303-105
kinematic_structure_schema	ISO 10303-105
management_resources_schema	ISO 10303-41
material_property_definition_schema	ISO 10303-45
material_property_representation_schema	ISO 10303-45
measure_schema	ISO 10303-41
method_definition_schema	ISO 10303-49
person_organization_schema	ISO 10303-41
presentation_appearance_schema	ISO 10303-46
presentation_definition_schema	ISO 10303-46
presentation_organization_schema	ISO 10303-46
presentation_resource_schema	ISO 10303-46
process_property_representation_schema	ISO 10303-49
process_property_schema	ISO 10303-49
product_concept_schema	ISO 10303-44
product_definition_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
product_property_representation_schema	ISO 10303-41
product_structure_schema	ISO 10303-44
qualified_measure_schema	ISO 10303-45
representation_schema	ISO 10303-43
security_classification_schema	ISO 10303-41
shape_aspect_definition_schema	ISO 10303-47
shape_dimension_schema	ISO 10303-47
shape_tolerance_schema	ISO 10303-47
support_resource_schema	ISO 10303-41
topology_schema	ISO 10303-42

5.2.1 Fundamental concepts and assumptions

This subclause describes fundamental concepts and assumptions related to the data structure defined within this part of ISO 10303.

ISO 10303-214 is designed to be used for data exchange, sharing, modeling, and storage in the design phase.

The intention is to gather information on a technical **product** related to various applications of product data.

5.2.1.1 Usage of global rules

Many of the relationships among different entities in the integrated resource parts of ISO 10303 are specified using the most generic cardinality of zero or more between two related entities. This cardinality means that the relationship is optional or there may be one or more instances of a related entity that is related to a single instance of the relating entity or vice versa.

This part of ISO 10303 uses global rules to constrain these cardinalities. Examples of these rules include **contract_requires_organization**, **approval_requires_approval_person_organization**, and **item_requires_version**.

Global rules are also used to restrict values of STRING type attributes to be only those that are applicable within the context of ISO 10303-214.

5.2.1.2 Various Views on Product Versions

Every **product** has to be related with one or more instances of **product_definition_formation** (enforced by the global rule **item_requires_version**), each representing a version of the **product**.

A **product_definition_formation** may be represented by different instances of **representation** (e.g. various kinds of geometric representations, kinematic structures, drawings), i.e. different points of view on a **product**.

Product_definition establishes the relationship between a specific **representation** and a **product_definition_formation**.

5.2.1.3 Relating the shape of a product to administrative product data

The shape of a **product** is represented by **shape_representation**. This entity and its subtypes are defining valid geometric and/or topologic combinations of entities to describe the **product**'s shape.

Shape_representation is related to **product_definition** by **shape_definition_representation** and **product_definition_shape**, which points to **product_definition** via two select types: **characterized_definition** and **characterized_product_definition**. Due to the global rule **shape_representation_subtype_exclusiveness** **shape_representation** shall be an instance of at most one of its subtypes.

5.2.1.4 Identification of a portion of product's shape

A portion of the whole **product**'s shape may be identified by a **shape_aspect**, that points to a **product_definition_shape**.

The **shape_aspect** may be related via **shape_definition_representation** to **shape_representation**, representing the portion of shape.

5.2.1.5 Types of shape representation

By this part of ISO 10303 nine model types for representation of **product**'s shape are defined:

- geometrically bounded 2D wireframe model represented by **geometrically_bounded_2d_wireframe_representation** (represented in the ARM by UoF G1);

- geometrically bounded 3D wireframe model represented by **geometrically_bounded_wireframe_shape_representation** (represented in the ARM by UoF G2);

- geometrically bounded surface model represented by **geometrically_bounded_surface_shape_representation** (represented in the ARM by UoF G8);

- connected surface model represented by **manifold_surface_shape_representation** (represented in the ARM by UoF G3);

- faceted Brep model represented by **faceted_brep_shape_representation** (represented in the ARM by UoF G4);

- Brep model represented by **advanced_brep_shape_representation** (represented in the ARM by UoF G5);

- compound Brep model represented by **compound_brep_shape_representation** and as special cases **edge_based_wireframe_shape_representation** and **non_manifold_surface_shape_representation** (represented in the ARM by UoF G6);

- CSG model represented by **csg_shape_representation** (represented in the ARM by UoF G7);

- hybrid model represented by **hybrid_3d_shape_representation** (included in the ARM in all geometric UoFs Gx).

All these types are subtypes of **shape_representation**. Each type is self contained, meaning that one type may not contain another, enforced by the global rule **shape_representation_subtype_exclusiveness**. Local rules, defined for every subtype, describe restrictions on geometric and/or topological entities, that may be referenced by the respective **shape_representation** subtype. The local rules in the subtypes of **shape_representation** uniformly allow for instances of **axis2_placement**, **mapped_item**, and one or more specific subtypes of **geometric_representation_item** according to the subtype of **shape_representation**. In the case of a **mapped_item** the mapped_representation of the associated instance of **representation_map** is restricted to the same subtype of **shape_representation** in order to avoid mixed shape representation types.

The **hybrid_3d_shape_representation** is an exception of this rule. It is especially created to allow for a mix of different kinds of **shape_representation** as it usually occurs in CAD

systems. Therefore the allowed types of instances of **geometric_representation_item** is not further constrained as well as the type the mapped_representation for a **mapped_item** instance is restricted to **shape_representation** only, but not to a specific subtype.

5.2.1.6 Relation of units, coordinate space, and accuracy information to shape information

Information on units related to the **product**'s shape as well as information on the coordinate space and the accuracy (uncertainty) of the shape information is assigned to instances of **shape_representation** by subtypes of **representation_context**, i.e. **global_unit_assigned_context**, **geometric_representation_context**, and **global_uncertainty_assigned_context**, respectively.

These subtypes allow for definition of a set of units, of the coordinate space dimension, and of one to three accuracy values for distance, angular, and curvature accuracy to be assigned to a **shape_representation**.

In the case where an instance of **representation_context** shall be shared among two or more **shape_representation** instances with different accuracy values complex instances of **shape_representation** and **uncertainty_assigned_representation** have to be used to assign the various accuracy values through the **uncertainty_assigned_representation** instances independent from the **representation_context**. The accuracy information defined by the **uncertainty_assigned_representation** takes precedence over any **global_uncertainty_assigned_context** that is referenced by the **shape_representation**.

5.2.1.7 Relating the shape of a component to the shape of its assembly

In order to build assembly structures, relationships between the assembly and component(s) have to be defined.

Two alternatives for the implementation of assembly structure representations are recommended:

- One alternative describes the assembly with the shape of the components built in. With this approach the shape of the component is mapped into the shape of the assembly via **mapped_item**;

- The other alternative describes the components and the assembly with their own shape information together with the transformation information between the the assembly and the component. This approach uses the **representation_relationship_with_transformation**;

Both mechanisms make sense even in mixed combinations. The product structure, i.e. **next_assembly_usage_occurence** is identical and is required for both approaches.

1) **mapped_item** approach:

This approach represents the assembled model completely. It is appropriate for explicit representation of assemblies. It uses **mapped_item** in combination with **repesentation_map** which takes a **representation**, i.e. the **shape_representation** of the component and turns it into

an item in a second **representation**, i.e. a **mapped_item** included in the **shape_representation** of the assembly. The transformation to be applied is determined from the mapping origin and the mapping target which are items of the two instances of **representation** (and therefore founded in the two contexts). **Mapped_item** is constrained such that:

- No **mapped_item** shall be dependent on itself to define the **representation** being mapped (acyclic);

- The mapping_origin shall be in the context of the **representation** referenced by **mapped_item**.mapping_source;

- The mapping_target shall be in the context of the representation including the **mapped_item**;

There is no constraint to prevent both instances of **representation** having the same context (i.e. the new item is shifted in position but does not change spaces). This allows the use of **mapped_item** to position sub-models in a single coordinate space. It is even possible to define an identity matrix transformation by referring to the same entity instance as both mapping_origin and mapping_target. This represents the case where a component is defined in (one of) its final positions. Until it is mapped into the second **representation**, it is not included in the **representation** even though defined in the same space.

2) **representation_relationship_with_transformation** approach:

This approach represents the components and the set of instructions how to build the assembly. This approach is adequate for the implicit representation of assemblies.

Thus, **representation_relationship_with_transformation** is used to define the relative positions within an assembly. It does not allow for:

- Inclusion: The decision to create a **representation** which brings all the components together is left to the receiving system;

- The ability to define a component in one position and then replicate it in the same space.

The **shape_representation** may be defined explicitly and completely for the assembly and the **representation_relationship_with_transformation** gives just the additional information how an instances of **shape_representation** for a component geometrically relates to the shape of the assembly.

The rule **coordinated_assembly_and_shape** implies that the relationships between the assembly and the component on the **product_definition** level and the **shape_representation** level has to be linked through a **context_dependent_shape_representation**. This is necessary to distinguish between several occurrences of the same component within an assembly.

According to ISO 10303-43 the attributes rep_1 and rep_2 of **representation_relationship_with_transformation** are determined as follows:

- rep_1 is defined as the **representation** with a context to which the transformation applies;

- rep_2 is the **representation** with a context which is the result of the transformation.

Based on this definition the attributes of **representation_relationship_with_transformation** should be instantiated as follows:

- rep_1 references the **shape_representation** for the **product_definition** that is the related_product_definition (identifying the component) of the **assembly_component_usage**;

- rep_2 references the **shape_representation** for the **product_definition** that is the relating_product_definition (identifying the assembly) of the **assembly_component_usage**;

Further restrictions are implied by the informal proposition on **representation_relationship_with_transformation**, defined in ISO 10303-43: If the transformation is an **item_defined_transformation**, the ordering of the instances of **representation** given for the inherited attributes of **representation_relationship** shall be consistent with the ordering of the two instances of **representation_item** given as attributes of **item_defined_transformation**.

Therefore the inherited attributes relating_representation_item and related_representation_item of **item_defined_transformation** have to be used as follows:

- related_representation_item references the **representation_item** (e.g. **axis2_placement_3d**) in a representation for the **product_definition** that is the related_product_definition (identifying the component) of the **assembly_component_usage**;

- relating_representation_item references the **representation_item** (e.g. **axis2_placement_3d**) in a representation for the **product_definition** that is the relating_product_definition (identifying the assembly) of the **assembly_component_usage**;

5.2.1.8 Representation of transformations

Transformation information is represented for an **item_defined_transformation** (inherited attributes relating_representation_item and related_representation_item) and for a **mapped_item** (attribute mapping_target of **mapped_item** and attribute mapping_origin of the referenced **representation_map**) as a combination of two instances of **representation_item**.

The allowed combinations for these two instances of **representation_item** are as follows:

- **axis2_placement_3d** for the related_representation_item or mapping_origin attribute and another instance of **axis2_placement_3d** for the relating_representation_item or mapping_target attribute;

- **axis2_placement_3d** for the related_representation_item or mapping_origin attribute and an instance of **cartesian_transformation_operator_3d** for the relating_representation_item or mapping_target attribute;

- **axis2_placement_2d** for the related_representation_item or mapping_origin attribute and another instance of **axis2_placement_2d** for the relating_representation_item or mapping_target attribute;

- **axis2_placement_2d** for the related_representation_item or mapping_origin attribute and an instance of **cartesian_transformation_operator_2d** for the relating_representation_item or mapping_target attribute.

For a **representation_relationship_with_transformation** there may be also an instance of **functionally_defined_transformation** used that is a **cartesian_transformation_operator**.

For any usage of an istance of **cartesian_transformation_operator** to define the transformation information of an instance of **assembly_component_usage** the derived attribute scl has to be 1.0, since scaling is not allowed components within an assembly.

5.2.1.9 Specification control

This clause describes the AIM entities used by ISO 10303-214 that provide the capability to describe automotive products with a large number of variants.

EXAMPLE 237 – Examples for automotive products are passenger cars, trucks, busses, engines, or other components of these products.

Because of the large number of variants there is no explicit representation for each of the variants that could be produced. The main concepts that are used to handle this large number of variants are the following:

- **product_class** is used to identify a set of similar products to be offered to the market.

A **product_class** is a **product_concept** and a **characterized_object**. The **characterized_object** may have **property_definition** entities associated to it through **property_definition**.definition and the select type **characterized_definition**.

The instances of **product_class** may build a hierarchical tree-like structure through instances of **product_concept_relationship** that have the attribute name set to 'class hierarchy' and that reference with the attribute related_product_concept the **product_class** that is a subclass of the **product_class** referenced by the attribute relating_product_concept. The supertypes **product_concept** and **characterized_object** are not intended to be instantiated on their own for this Part of ISO 10303. This is enforced by the rules **subtype_mandatory_product_concept** and **subtype_mandatory_characterized_object**.

- **product_concept_feature** is used to describe a characteristic of the products within a **product_class**.

The **product_concept_feature** is associated to a **product_class** through a **product_concept_feature_association** with the attribute name set to 'replaceable standard', 'non replaceable standard', 'availability', 'identification', 'option', or any other non standardized value. For a definition of the meaning of the standardized values see 4.2.68 or clause 5.2 (**product_concept_feature_association** imported entity modification for permissive list definition).

A **package_product_concept_feature** is a subtype of **product_concept_feature** that is used when the characteristic is in fact a set of other characteristics, e.g. a package of several options for a passenger car. In this case there is a **conditional_concept_feature** with the inherited attribute name set to 'inclusion' and the attribute condition referencing a **concept_feature_relationship_with_condition** with the attribute conditional_operator set to 'implication'. The inherited attribute relating_product_concept_feature of the **concept_feature_relationship_with_condition** references the **package_product_concept_feature** and the inherited attribute related_product_concept_feature references either the one **product_concept_feature** that belongs to the package or another **conditional_concept_feature** that references the **product_concept_feature** of the package through an 'and' operation.

– **product_concept_feature_category** is used to group similar instances of **product_concept_feature**, e.g. the different outside colour characteristics of the products.

A **product_concept_feature_category** is a **group** that groups instances of **product_concept_feature**. Instances of **product_concept_feature** are assigned to **product_concept_feature_category** through **group_member_assignment** that is a subtype of **group_assignment**. For this **group_assignment** the attribute items references through the select type **group_member_item** instances of **product_concept_feature** or **package_product_concept_feature**, the attribute assigned_group references **product_concept_feature_category** or **exclusive_product_concept_feature_category**, the attribute role references **group_role** with the attribute name set to 'specification category member'.

Exclusive_product_concept_feature_category is subtype of **product_concept_feature_category** that is used when the members of the **group** are mutually exclusive in their usage within a product, e.g. there is exactly one outside colour that can be choosen for a particular product.

Product_concept_feature_category is associated to **product_class** through **product_concept_feature_category_usage** that is subtype of **group_assignment**.

For this **group_assignment** the attribute items references through the select type **category_usage_item** an instance of **product_class**, the attribute assigned_group references **product_concept_feature_category** or **exclusive_product_concept_feature_category**, the attribute role references **group_role** with the attribute name set to 'mandatory category usage' or 'optional category usage'. For a definition of the meaning of the two values see 4.2.64 or clause 5.2 (**product_concept_feature_category_usage**, where rule explanation for allowed values of **group_role**.name).

– **conditional_concept_feature** is used to control the usage of a **product** (in the ARM: an item classified as 'part') within the products of a **product_class** and to represent conditions for instances of **product_class**.

The **conditional_concept_feature** is subtype of **product_concept_feature** with the attribute name set to 'expression' or 'inclusion'. The **conditional_concept_feature** is used to build up boolean conditions for instances of **product_concept_feature**. The **conditional_concept_feature** references with its attribute condition a **concept_feature_relationship_with_condition** that represents a binary boolean operation. A **concept_feature_relationship_with_condition** is subtype of **concept_feature_relationship** with the attribute conditional_operator set to 'and', 'or', 'oneof', 'not', or 'implication'. For a definition of the meaning of the values see 4.2.417 or clause 5.2 (**concept_feature_relationship_with_condition** imported entity modification for permissive list values for attribute conditional_operator).

Conditional_concept_feature may be associated to a **product_class** through **product_concept_feature_association** with the attribute name set to 'part_usage', 'identification', 'validity', or any other non standardized value. For a definition of the meaning of the standardized values see4.2.65 or clause 5.2 (**product_concept_feature_association** imported entity modification for permissive list definition).

The value 'part usage' indicates that the **conditional_concept_feature** controls the usage of a part within a product of the associated **product_class**. In this case the **product_concept_feature_association** is referenced by a **configured_effective_context_assignment** that

is a subtype of **effective_context_assignment** subtype of **effectivity_assignment** through the select type **configured_effective_context_item**. The **configured_effective_context_assignment** is instatiated as a complex instance with **configured_effective_item_assignment** that is a subtype of **effective_item_assignment** subtype of **effectivity_assignment**.

The **configured_effective_item_assignment** references through the select type **configured_effective_item** the instance of **solution_based_product_definition** that identifies the variant, i.e. the **product** to be controlled.

The complex instance references through the inherited attribute assigned_effectivity an instance of **effectivity** that controls the part usagen in terms of start and end date (i.e. **dated_effectivity**), lot identification (i.e. **lot_effectivity**), serial number range (i.e. **serial_numbered_effectivity**), or start and end events (i.e. **event_based_effectivity**), like 'start of production'.

- **product_definition** that has a **product_definition_context** with the name attribute set to 'functional definition' is used to describe the functional requirements for products in a **product_class** and for instances of **product_component**.

Such instances of **product_definition** for functional requirements may be related to instances of **product_class** through an instance of **configuration_design** and **configuration_item**. In this case the name attribute of **configuration_item** is set to 'class function definition' and the name attribute of **configuration_design** is set to 'possibility' or 'main function'. For a definition of the meaning of the standardized values see4.2.66 or clause 5.2 (**configuration_design** imported entity modification for permissive list definition). In this case one instance of **configuration_item** reflects all the main and possible functions of a **product_class**.

To relate the functional requirements represented by instances of **product_definition** to an instance of **product_component** another instance of **configuration_design** is used with the name attribute set to 'component function definition'. In this case the **configuration_item** is a **product_component** and the functional requirement does not apply globally to the products of the **product_class**, but to one identified **product_component** of that **product_class**.

- **product_component** is used to describe the common decomposition structure of all products of a **product_class**.

A **product_component** is a **configuration_item_with_context** and a **characterized_object**. The **characterized_object** may have **property_definition** entities associated to it through **property_definition**.definition and the select type **characterized_definition**.

The instances of **product_component** may build a hierarchical tree-like structure through instances of **configuration_item_relationship** that have the attribute name set to 'decomposition' and that reference with the attribute related_configuration_item the **product_component** that is a subcomponent of the **product_component** referenced by the attribute relating_configuration_item. The supertypes **configuration_item_with_context** and **characterized_object** are not intended to be instantiated on their own for this Part of ISO 10303. This is enforced by the rules **subtype_mandatory_configuration_item_with_context** and **subtype_mandatory_characterized_object**.

- **solution_based_product_definition** is used to describe the variants for a **product_component**.

A **solution_based_product_definition** is a **product_definition**. An instance of **solution_based_product_definition** may be associated to the **product_component** where it is a vari-

ant of through an instance of **configuration_design** with the name attribute set to 'solution alternate definition'.

An instance of **solution_based_product_definition** may also be a variant of another instance of **solution_based_product_definition**, e.g. where the first level defines technically different solutions and the second level defines different suppliers for the same technical solution. In this case an instance of **product_definition_relationship** with the name attribute set to 'solution alternative definition' is used to relate (as related_product_definition) the first **solution_based_product_definition** instance to the second **solution_based_product_definition** instance (as relating_product_definition) that is the basis for the first one.

All instances of **solution_based_product_definition** that are defined through this path as variant of the same instance of **product_component** or another instance of **solution_based_product_definition** are the mutually exclusive alternate solutions (variants) on this basis.

- **product_definition_occurrence** is used to identify elements of a **solution_based_product_definition**, i.e. the elements of a particular variant for a **product_component**.

A **product_definition_occurrence** is a **product_definition** that has a **product_definition_context** with the name attribute set to 'definitional occurrence'.

A **product_definition_occurrence** defines exactly one occurrence of a part within the assembly structure of a product. An instance of **product_definition_occurrence** may be defined as an element (as related_product_definition) of a **solution_based_product_definition** instance (as relating_product_definition) through a **product_defninition_relationship** with the name attribute set to 'solution instance association'.

5.2.1.10 Form Features

This clause describes the concept concerning the definition and application of form features in this part of ISO 10303.

This standard includes the definition of various form features for the application in panel like parts (i.e. with negligible thickness) and for the application in solid volume parts. Also included is a concept for so-called general features whose semantics and parameters are not predefined as well as for transition features and for threads. This standard distinguishes between the definition of the features that is realized by the entity **feature_definition** which is a subtype of **characterized_object** and their occurrence on a part.

Predefined form features specified for an application in panels are **bead**, **joggle**, **hole_in_panel** that may be a bearing **hole**, **locator** and **feature_in_panel** that is either a boss, a pocket, or a **stair_step**. These features are provided with a set of parameters, profiles, or paths that are supposed to be sufficient to determine the shape of a part after the feature is applied.

Predefined form features specified for an application in solid parts are **boss**, **pocket**, **rib**, **round_hole** and **slot**.

These features are specified by a volume that is defined by a profile and by a path along which the profile is swept. The feature itself is created by adding or subtracting this volume from the part's shape.

For the definition of predefined features a set of predefined profiles, i.e. **circular_closed_profile**,

ngon_closed_profile, **rectangular_closed_profile**, **partial_circular_profile**, **rounded_u_-profile**, **square_u_profile**, **tee_profile**, and **vee_profile**, as well as two general profiles (**closed_-path_profile** and **open_path_profile**) are included. The paths that may be used to define form features are represented by the entity **path_feature_component**. This is either a predefined type of path (**path_feature_component**.name = 'linear', 'partial circular', or 'complete circular') that is specified by a set of parameters or a general type of path (**path_feature_component**.name = 'complex') that is specified by **curve**.

Paths and profiles are specified as instances of **shape_aspect**. Here, a distinction is made between the definition of such feature components and their occurrence. The definitions of paths and profiles are subtypes of **shape_aspect** that reference a **feature_component_definition** which is a subtype of **characterized_object** as 'of_shape'.

This definition of a feature component is associated with its occurrence which is a **shape_aspect** of a **feature_definition** through a subtype of **shape_aspect_relationship** named **shape_defining_relationship**.

For the application of form feature definitions this standard supports two approaches:

- by the entity **instanced_feature** that is a **shape_aspect** of the **product_definition_-shape** and at the same time a subtype of **feature_definition**;

- by the entity **placed_feature** that is also a **shape_aspect** of the **product_definition_-shape** and related to the **feature_definition** by an instance of **mapped_item** and **representation_map**;

The approach using **instanced_feature** supports the identification of 'areas of interest' on an already designed shape that are of a certain meaning to the user, e.g. that correspond to a process operation. The identification of parameters on these features does not mean that a variation of the parameter values results in an (automatic) update of the shape but is only meant to support the specification of these areas in a more appropriate way, e.g. to support process planning. In the case of the application of a feature as an **instanced_feature** the feature itself as well as all of its components (paths or profiles) are defined in the part coordinate system.

The approach using **placed_feature** supports the definition of the feature independent from its application in its own coordinate system and a placement of the **representation** of the **feature_definition** into the **representation** of the **placed_feature** using a **mapped_item** and a **representation_map**. The shape of the feature needs not to be represented since a receiving system is supposed to be able to create the shape that results from the application of the **feature_definition** on a base shape interpreting the values of the feature parameters and the semantics included in the type of feature. In the case of the application of a feature as a **placed_feature** the feature itself and possibly a path used for the feature definition are defined in one coordinate system, a used profile is defined in its own coordinate system. Profiles are placed in the feature coordinate system using a **mapped_item** and a **representation_map**.

This standard supports the application of surface conditions and tolerances to form features only implicitly by applying surface conditions and tolerances to elements of the shape representation of the form feature. An exception are tolerances applied to the location and orientation of the features that may be realized by using **axis2_placement** entities representing the feature coordinate system.

5.2.2 automotive_design types

5.2.2.1 action_item

An **action_item** type specifies those objects to which an **action** may be assigned.

EXPRESS specification:

```
*)
TYPE action_item = SELECT
  (action,
   action_method,
   action_property,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   conditional_concept_feature,
   configurable_item,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   document,
   draughting_model,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   project,
   property_definition,
   representation,
   shape_representation,
   solution_based_product_definition);
END_TYPE;
(*
```

5.2.2.2 action_request_item

An **action_request_item** type specifies those objects to which a **versioned_action_request** may be assigned.

EXPRESS specification:

```
*)
TYPE action_request_item = SELECT
  (action_method,
   action_property,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   conditional_concept_feature,
   configurable_item,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   document,
   draughting_model,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   property_definition,
   representation,
   shape_representation,
   solution_based_product_definition);
END_TYPE;
(*
```

5.2.2.3 approval_item

An **approval_item** type specifies those objects to which an **approval** may be assigned.

EXPRESS specification:

```
*)
TYPE approval_item = SELECT
  (action,
   action_directive,
   alternate_product_relationship,
   applied_action_assignment,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   conditional_concept_feature,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   document,
   draughting_model,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   project,
   representation,
   shape_representation,
   solution_based_product_definition);
END_TYPE;
(*
```

5.2.2.4 category_usage_item

A **category_usage_item** type specifies those objects to which a **group** may be assigned.

EXPRESS specification:

```
*)
TYPE category_usage_item = SELECT
  (product_class);
END_TYPE;
```

```
(*
```

5.2.2.5 certification_item

A **certification_item** type specifies those objects to which a **certification** may be assigned.

EXPRESS specification:

```
*)
TYPE certification_item = SELECT
  (solution_based_product_definition);
END_TYPE;
(*
```

5.2.2.6 configured_effective_context_item

A **configured_effective_context_item** type specifies those objects to which an **effectivity** may be assigned. The **configured_effective_context_item** provides the context in which the assigned **effectivity** is valid.

EXPRESS specification:

```
*)
TYPE configured_effective_context_item = SELECT
  (product_concept_feature_association);
END_TYPE;
(*
```

5.2.2.7 configured_effective_item

A **configured_effective_item** type specifies those objects to which an **effectivity** may be assigned. The **effectivity** specifies the valid uses of the **configured_effective_item** for configuration purposes.

EXPRESS specification:

```
*)
TYPE configured_effective_item = SELECT
  (action,
   solution_based_product_definition);
END_TYPE;
(*
```

5.2.2.8 contract_item

A **contract_item** type specifies those objects to which a **contract** may be assigned.

EXPRESS specification:

```
*)
TYPE contract_item = SELECT
  (action,
   product_definition_formation);
END_TYPE;
(*
```

5.2.2.9 date_and_organization_item

A **date_and_organization_item** type specifies those objects to which an **organization** and a **date** or **date_and_time** may be assigned.

EXPRESS specification:

```
*)
TYPE date_and_organization_item = SELECT
  (action,
   action_property,
   applied_action_assignment,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   conditional_concept_feature,
   configurable_item,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   document,
   draughting_model,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   project,
```

```
   property_definition,
   representation,
   shape_representation,
   solution_based_product_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.10 date_and_person_and_organization_item

A **date_and_person_and_organization_item** type specifies those objects to which a **person_-and_organization** and a **date** or **date_and_time** may be assigned.

EXPRESS specification:

```
*)
TYPE date_and_person_and_organization_item = SELECT
  (action,
   applied_action_assignment,
   applied_organization_assignment,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   conditional_concept_feature,
   configurable_item,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   document,
   draughting_model,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   project,
   property_definition,
```

```
   representation,
   shape_representation,
   solution_based_product_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.11 date_and_person_item

A **date_and_person_item** type specifies those objects to which a **person** and a **date** or **date_-and_time** may be assigned.

EXPRESS specification:

```
*)
TYPE date_and_person_item = SELECT
  (action,
   action_property,
   applied_action_assignment,
   applied_organization_assignment,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   conditional_concept_feature,
   configurable_item,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   document,
   draughting_model,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   project,
   property_definition,
```

```
   representation,
   shape_representation,
   solution_based_product_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.12 date_and_time_item

A **date_and_time_item** type specifies those objects to which a **date_and_time** may be assigned.

EXPRESS specification:

```
*)
TYPE date_and_time_item = SELECT
  (action,
   action_property,
   applied_action_assignment,
   applied_date_and_organization_assignment,
   applied_date_and_person_and_organization_assignment,
   applied_date_and_person_assignment,
   approval_person_organization,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   certification,
   conditional_concept_feature,
   configurable_item,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   document,
   draughting_model,
   event_based_effectivity,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
```

```
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   project,
   property_definition,
   representation,
   retention,
   shape_representation,
   solution_based_product_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.13 date_item

A **date_item** type specifies those objects to which a **date** may be assigned.

EXPRESS specification:

```
*)
TYPE date_item = SELECT
  (action,
   action_property,
   applied_action_assignment,
   applied_date_and_organization_assignment,
   applied_date_and_person_and_organization_assignment,
   applied_date_and_person_assignment,
   approval_person_organization,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   certification,
   conditional_concept_feature,
   configurable_item,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   document,
   draughting_model,
   event_based_effectivity,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
```

```
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   project,
   property_definition,
   representation,
   retention,
   shape_representation,
   solution_based_product_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.14 digital_document_reference_item

A **digital_document_reference_item** type specifies those objects to which a digital **document** may be assigned.

EXPRESS specification:

```
*)
TYPE digital_document_reference_item = SELECT
  (externally_defined_representation);
END_TYPE;
(*
```

5.2.2.15 document_reference_item

A **document_reference_item** type specifies those objects to which a **document** may be assigned.

EXPRESS specification:

```
*)
TYPE document_reference_item = SELECT
  (action,
   action_directive,
   action_property,
   action_relationship,
   approval,
   assembly_component_usage,
   characterized_product_category,
   classification_system,
   configurable_item,
   configuration_design,
```

```
   configuration_item_relationship,
   contract,
   drawing_sheet_revision,
   exclusive_product_concept_feature_category,
   externally_defined_thread,
   material_designation,
   mechanical_design_geometric_presentation_area,
   package_product_concept_feature,
   presentation_area,
   process_plan,
   product_category,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_category,
   product_definition,
   product_definition_formation_relationship,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   project,
   property_definition,
   representation,
   resource_requirement_type,
   retention,
   shape_aspect_relationship,
   solution_based_product_definition);
END_TYPE;
(*
```

5.2.2.16 effective_context_item

An **effective_context_item** specifies those objects to which an **effectivity** may be assigned. The **effective_context_item** provides the context in which the assigned **effectivity** is valid.

EXPRESS specification:

```
*)
TYPE effective_context_item = SELECT
  (organization,
   product_definition_context);
END_TYPE;
(*
```

5.2.2.17 effective_item

An **effective_item** type specifies those objects to which an **effectivity** may be assigned. The **effectivity** specifies the valid use of the **effective_item**.

EXPRESS specification:

```
*)
TYPE effective_item = SELECT
  (action_property,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   conditional_concept_feature,
   configurable_item,
   configuration_design,
   document,
   draughting_model,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   property_definition,
   representation,
   security_classification,
   shape_representation,
   solution_based_product_definition);
END_TYPE;
(*
```

5.2.2.18 element_delivery_item

An **element_delivery_item** type specifies those objects to which an **element_delivery** may be assigned.

EXPRESS specification:

```
*)
TYPE element_delivery_item = SELECT
  (applied_action_assignment);
END_TYPE;
```

```
(*
```

5.2.2.19 event_context_item

An **event_context_item** type specifies those objects to which an **event_reference** may be assigned. The **event_context_item** specifies the context to which the **event_reference** applies.

EXPRESS specification:

```
*)
TYPE event_context_item = SELECT
  (action,
   action_property,
   applied_action_assignment,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   conditional_concept_feature,
   configurable_item,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   document,
   draughting_model,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   project,
   property_definition,
   representation,
   shape_representation,
   solution_based_product_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.20 event_item

An **event_item** type specifies those objects to which an **event_reference** may be assigned.

EXPRESS specification:

```
*)
TYPE event_item = SELECT
  (event_based_effectivity,
   project);
END_TYPE;
(*
```

5.2.2.21 external_identification_item

An **external_identification_item** type specifies those objects to which an external identifier and the external source for which the identifier is valid may be assigned.

EXPRESS specification:

```
*)
TYPE external_identification_item = SELECT
  (document);
END_TYPE;
(*
```

5.2.2.22 group_item

A **group_item** type specifies those objects to which a **group** may be assigned.

EXPRESS specification:

```
*)
TYPE group_item = SELECT
  (product_component);
END_TYPE;
(*
```

5.2.2.23 group_member_item

A **group_member_item** type specifies those objects to which a **group** may be assigned in a way that the group_member_item is a member of the **group**.

EXPRESS specification:

```
*)
```

```
TYPE group_member_item = SELECT
  (approval_status,
   characterized_product_category,
   document_type,
   drawing_definition,
   product_concept_feature,
   product_concept_feature_category_usage,
   product_definition,
   representation_item,
   security_classification_level,
   shape_aspect);
END_TYPE;
(*
```

5.2.2.24 identification_item

An **identification_item** type specifies those objects to which an **identifier** may be assigned.

EXPRESS specification:

```
*)
TYPE identification_item = SELECT
  (approval_status,
   assembly_component_usage,
   classification_system,
   document,
   document_type,
   draughting_model,
   externally_defined_classification_system,
   measure_representation_item,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_category,
   product_definition,
   product_definition_occurrence,
   security_classification_level,
   shape_aspect_relationship,
   shape_representation,
   solution_based_product_definition,
   version_assignment);
END_TYPE;
(*
```

5.2.2.25 multi_language_attribute_item

A **multi_language_attribute_item** type specifies those objects to which an alternate attribute value specified in an additional language for an existing attribute may be assigned.

EXPRESS specification:

```
*)
TYPE multi_language_attribute_item = SELECT
  (action,
   action_directive,
   action_method,
   action_property,
   action_relationship,
   action_resource_requirement,
   alternate_product_relationship,
   application_context,
   approval,
   area_presentation_relationship,
   assembly_component_usage,
   assembly_component_usage_substitute,
   camera_image_2d_with_scale,
   camera_image_3d_with_scale,
   certification,
   characterized_product_category,
   classification_system,
   configurable_item,
   configuration_design,
   configuration_item_relationship,
   constructive_geometry_representation_relationship,
   data_environment,
   default_tolerance_table_cell,
   descriptive_representation_item,
   dimensional_location,
   dimensional_size,
   document,
   document_relationship,
   draughting_model,
   draughting_title,
   drawing_sheet_revision,
   effectivity,
   effectivity_relationship,
   externally_defined_classification_attribute,
   externally_defined_classification_system,
   externally_defined_product_category,
   externally_defined_representation,
   geometric_tolerance,
   group,
   group_relationship,
   identification_role,
   make_from_usage_option,
   material_property,
   measure_representation_item,
   mechanical_design_geometric_presentation_representation,
   model_change,
   organization_relationship,
   organization_role,
```

```
   pair_actuator,
   presentation_area,
   presentation_layer_assignment,
   presentation_view,
   presentation_view_relationship,
   process_change,
   process_plan,
   process_product_association,
   product,
   product_category,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_definition,
   product_definition_formation,
   product_definition_formation_relationship,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_relationship ,
   product_definition_shape,
   product_definition_substitute,
   project,
   property_change,
   property_definition,
   property_definition_relationship,
   property_definition_representation,
   rep_item_group,
   representation,
   representation_item,
   representation_relationship,
   resource_requirement_type,
   retention,
   security_classification,
   shape_aspect,
   shape_aspect_relationship,
   shape_dimension_representation,
   shape_representation,
   solution_based_product_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.26 organization_item

An **organization_item** type specifies those objects to which an **organization** may be assigned.

EXPRESS specification:

```
*)
```

```
TYPE organization_item = SELECT
  (action,
   action_property,
   applied_action_assignment,
   applied_identification_assignment,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   conditional_concept_feature,
   configurable_item,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   contract,
   document,
   draughting_model,
   element_delivery,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   project,
   property_definition,
   representation,
   shape_representation,
   solution_based_product_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.27 person_and_organization_item

A **person_and_organization_item** type specifies those objects to which a **person_and_-organization** may be assigned.

EXPRESS specification:

```
*)
TYPE person_and_organization_item = SELECT
  (action,
   action_property,
   applied_action_assignment,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   conditional_concept_feature,
   configurable_item,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   document,
   draughting_model,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   project,
   property_definition,
   representation,
   shape_representation,
   solution_based_product_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.28 person_item

A **person_item** type specifies those objects to which a **person** may be assigned.

EXPRESS specification:

```
*)
```

```
TYPE person_item = SELECT
  (action,
   action_property,
   applied_action_assignment,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   conditional_concept_feature,
   configurable_item,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   document,
   draughting_model,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   project,
   property_definition,
   representation,
   shape_representation,
   solution_based_product_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.29 presented_item_select

A **presented_item_select** type specifies those objects which may be presented by a drawing or in a presentation view.

EXPRESS specification:

```
*)
TYPE presented_item_select = SELECT
```

```
  (action,
   assembly_component_usage,
   configuration_design,
   exclusive_product_concept_feature_category,
   package_product_concept_feature,
   process_plan,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_category,
   product_definition,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   representation,
   solution_based_product_definition);
END_TYPE;
(*
```

5.2.2.30 process_plan_item

A **process_plan_item** type specifies those objects to which a **process_plan** may be assigned.

EXPRESS specification:

```
*)
TYPE process_plan_item = SELECT
  (process_product_association);
END_TYPE;
(*
```

5.2.2.31 project_item

A **project_item** type specifies those objects to which a **project** may be assigned.

EXPRESS specification:

```
*)
TYPE project_item = SELECT
  (product_class);
END_TYPE;
(*
```

5.2.2.32 retention_item

A **retention_item** type specifies those objects to which a **retention** may be assigned.

EXPRESS specification:

```
*)
TYPE retention_item = SELECT
  (action,
   action_property,
   applied_action_assignment,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   conditional_concept_feature,
   configurable_item,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   document,
   draughting_model,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_association,
   product_concept_feature_category,
   product_concept_feature_category_usage,
   product_definition,
   product_definition_formation,
   product_definition_occurrence,
   product_definition_relationship,
   product_definition_substitute,
   project,
   property_definition,
   representation,
   shape_representation,
   solution_based_product_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.33 security_classification_item

A **security_classification_item** type specifies those objects to which a **security_classification** may be assigned.

EXPRESS specification:

```
*)
TYPE security_classification_item = SELECT
  (action,
   action_directive,
   action_property,
   applied_action_assignment,
   assembly_component_usage,
   assembly_component_usage_substitute,
   auto_design_drawing_revision,
   configuration_design,
   configured_effective_context_assignment,
   configured_effective_item_assignment,
   document,
   draughting_model,
   kinematic_property_definition,
   material_designation,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product,
   product_class,
   product_component,
   product_concept_feature,
   product_concept_feature_category,
   product_definition,
   product_definition_formation,
   project,
   property_definition,
   representation,
   shape_representation,
   solution_based_product_definition,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.34 specified_item

A **specified_item** type specifies those objects to which a **draughting_specification_reference** may be assigned.

EXPRESS specification:

```
*)
TYPE specified_item = SELECT
  (drawing_revision);
END_TYPE;
(*
```

5.2.2.35 version_item

A **version_item** type specifies those objects to which a version identifier may be assigned.

EXPRESS specification:

```
*)
TYPE version_item = SELECT
  (action_directive,
   configurable_item,
   document,
   draughting_model,
   mechanical_design_geometric_presentation_representation,
   physically_modelled_product_definition,
   presentation_area,
   process_plan,
   product_class,
   product_component,
   product_concept_feature,
   product_definition,
   representation,
   shape_representation,
   solution_based_product_definition);
END_TYPE;
(*
```

5.2.3 automotive_design entities

5.2.3.1 automotive_design entity definitions

5.2.3.1.1 applied_action_assignment

An **applied_action_assignment** is a type of **action_assignment** that assigns an **action** to an **action_item.**

EXPRESS specification:

```
*)
ENTITY applied_action_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF action_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **action** is assigned.

5.2.3.1.2 applied_action_request_assignment

An **applied_action_request_assignment** is a type of **action_request_assignment** that assigns a **versioned_action_request** to an **action_request_item.**

EXPRESS specification:

```
*)
ENTITY applied_action_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF action_request_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **versioned_action_request** is assigned.

5.2.3.1.3 applied_approval_assignment

An **applied_approval_assignment** is a type of **approval_assignment** that assigns an **approval** to an **approval_item.**

EXPRESS specification:

```
*)
ENTITY applied_approval_assignment
  SUBTYPE OF (approval_assignment);
  items : SET [1:?] OF approval_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **approval** is assigned.

5.2.3.1.4 applied_certification_assignment

An **applied_certification_assignment** is a type of **certification_assignment** that assigns a **certification** to a **certification_item.**

EXPRESS specification:

```
*)
ENTITY applied_certification_assignment
  SUBTYPE OF (certification_assignment);
```

```
  items : SET [1:?] OF certification_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **certification** is assigned.

5.2.3.1.5 applied_contract_assignment

An **applied_contract_assignment** is a type of **contract_assignment** that assigns a **contract** to a **contract_item.**

EXPRESS specification:

```
*)
ENTITY applied_contract_assignment
  SUBTYPE OF (contract_assignment);
  items : SET [1:?] OF contract_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **contract** is assigned.

5.2.3.1.6 applied_date_and_organization_assignment

An **applied_date_and_organization_assignment** is a type of **organization_assignment** that assigns an **organization** and a **date** or **date_and_time** to a **date_and_organization_-item.**

EXPRESS specification:

```
*)
ENTITY applied_date_and_organization_assignment
  SUBTYPE OF (organization_assignment);
  items : SET [1:?] OF date_and_organization_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **organization** is assigned.

5.2.3.1.7 applied_date_and_person_and_organization_assignment

An **applied_date_and_person_and_organization_assignment** is a type of **person_and_organization_assignment** that assigns a **person_and_organization** and a **date** or **date_and_time** to an **date_and_person_and_organization_item.**

EXPRESS specification:

```
*)
ENTITY applied_date_and_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
  items : SET [1:?] OF date_and_person_and_organization_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **person_and_organization** is assigned.

5.2.3.1.8 applied_date_and_person_assignment

An **applied_date_and_person_assignment** is a type of **person_assignment** that assigns a person and a **date** or **date_and_time** to a **date_and_person_item.**

EXPRESS specification:

```
*)
ENTITY applied_date_and_person_assignment
  SUBTYPE OF (person_assignment);
  items : SET [1:?] OF date_and_person_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **person** is assigned.

5.2.3.1.9 applied_date_and_time_assignment

An **applied_date_and_time_assignment** is a type of **date_and_time_assignment** that assigns a **date_and_time** to a **date_and_time_item.**

EXPRESS specification:

```
*)
```

```
ENTITY applied_date_and_time_assignment
  SUBTYPE OF (date_and_time_assignment);
  items : SET [1:?] OF date_and_time_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **date_and_time** is assigned.

5.2.3.1.10 applied_date_assignment

An **applied_date_assignment** is a type of **date_assignment** that assigns a **date** to a **date_-item.**

EXPRESS specification:

```
*)
ENTITY applied_date_assignment
  SUBTYPE OF (date_assignment);
  items : SET [1:?] OF date_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **date** is assigned.

5.2.3.1.11 applied_document_reference

An **applied_document_reference** is a type of **document_reference** that assigns a **document** to a **document_reference_item.**

EXPRESS specification:

```
*)
ENTITY applied_document_reference
  SUBTYPE OF (document_reference);
  items : SET [1:?] OF document_reference_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **document** is assigned.

5.2.3.1.12 applied_effective_context_assignment

An **applied_effective_context_assignment** is a type of **effective_context_assignment** that assigns an **effectivity** to an **effective_context_item.**

EXPRESS specification:

```
*)
ENTITY applied_effective_context_assignment
  SUBTYPE OF (effective_context_assignment);
  items : SET [1:?] OF effective_context_item;
WHERE
  WR1:  SIZEOF(['AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY',
         'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY']
               * TYPEOF(SELF.assigned_effectivity)) = 1;
  WR2:  (SIZEOF( QUERY( i <* SELF.items |
          'AUTOMOTIVE_DESIGN.ORGANIZATION' IN TYPEOF(i)
         )) = 0) OR
        (SELF.role.name = 'concerned organization');
  WR3:  (SIZEOF( QUERY( i <* SELF.items |
         'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_CONTEXT' IN TYPEOF(i)
         )) = 0) OR
        (SELF.role.name = 'effectivity context');
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **effectivity** is assigned.

Formal propositions:

WR1: The **applied_effective_context_assignment** shall reference as its **assigned_effectivity** an **effectivity** that is of type **dated_effectivity** or **event_based_effectivity**.

WR2: If the **applied_effective_context_assignment** contains an **organization** in its set of **items**, it shall reference as its **role** an **effectivity_role** that has a **name** of 'concerned organization'.

WR3: If the **applied_effective_context_assignment** contains a **product_definition_context** in its set of **items**, it shall reference as its **role** an **effectivity_role** that has a **name** of 'effectivity context'.

5.2.3.1.13 applied_effective_item_assignment

An **applied_effective_item_assignment** is a type of **effective_item_assignment** that assigns an **effectivity** to an **effective_item.**

EXPRESS specification:

```
*)
ENTITY applied_effective_item_assignment
  SUBTYPE OF (effective_item_assignment);
  items : SET [1:?] OF effective_item;
WHERE
  WR1:  SIZEOF(['AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY',
         'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY'] *
               TYPEOF(SELF.assigned_effectivity)) = 1;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **effectivity** is assigned.

Formal propositions:

WR1: The **applied_effective_item_assignment** shall reference as its **assigned_effectivity** an **effectivity** that is of type **dated_effectivity** or **event_based_effectivity**.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **applied_effective_item_assignment** entity:

– product_definition_substitute_requires_effectivity_assignment (see 5.2.4.68).

5.2.3.1.14 applied_external_identification_assignment

An **applied_external_identification_assignment** is a type of **external_identification_assignment** that assigns an external identifier and an external source to an **external_identification_item.**

EXPRESS specification:

```
*)
ENTITY applied_external_identification_assignment
  SUBTYPE OF (external_identification_assignment);
  items : SET [1:?] OF external_identification_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **identifier** is assigned.

5.2.3.1.15 applied_group_assignment

An **applied_group_assignment** is a type of **group_assignment** that assigns a **group** to a **group_item.**

EXPRESS specification:

```
*)
ENTITY applied_group_assignment
  SUBTYPE OF (group_assignment);
  items : SET [1:?] OF group_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **group** is assigned.

5.2.3.1.16 applied_identification_assignment

An **applied_identification_assignment** is a type of **identification_assignment** that assigns an **identifier** to an **identification_item.**

EXPRESS specification:

```
*)
ENTITY applied_identification_assignment
  SUBTYPE OF (identification_assignment);
  items : SET [1:?] OF identification_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **identifier** is assigned.

5.2.3.1.17 applied_multi_language_attribute_assignment

An **applied_multi_language_assignment** is a type of **multi_language_attribute_assignment** that assigns an alternate **label** to a **multi_language_attribute_item.**

EXPRESS specification:

```
*)
ENTITY applied_multi_language_attribute_assignment
  SUBTYPE OF (multi_language_attribute_assignment);
```

```
  items : SET [1:?] OF multi_language_attribute_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **label** is assigned.

5.2.3.1.18 applied_organization_assignment

An **applied_organization_assignment** is a type of **organization_assignment** that assigns an **organization** to an **organization_item.**

EXPRESS specification:

```
*)
ENTITY applied_organization_assignment
  SUBTYPE OF (organization_assignment);
  items : SET [1:?] OF organization_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **organization** is assigned.

5.2.3.1.19 applied_person_and_organization_assignment

An **applied_person_and_organization_assignment** is a type of **person_and_organization_-assignment** that assigns a **person_and_organization** to a **person_and_organization_item.**

EXPRESS specification:

```
*)
ENTITY applied_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
  items : SET [1:?] OF person_and_organization_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **person_and_organization** is assigned.

5.2.3.1.20 applied_person_assignment

An **applied_person_assignment** is a type of **person_assignment** that assigns a **person** to a **person_item.**

EXPRESS specification:

```
*)
ENTITY applied_person_assignment
  SUBTYPE OF (person_assignment);
  items : SET [1:?] OF person_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **person** is assigned.

5.2.3.1.21 applied_presented_item

An **applied_presented_item** is a type of **presented_item** that assigns a **presentation_representation,** or a **presentation_set** to an item in the **presented_item_select.**

EXPRESS specification:

```
*)
ENTITY applied_presented_item
  SUBTYPE OF (presented_item);
  items : SET [1:?] OF presented_item_select;
END_ENTITY;
(*
```

5.2.3.1.22 applied_security_classification_assignment

An **applied_security_classification_assignment** is a type of **security_classification_assignment** that assigns a **security_classification** to a **security_classification_item.**

EXPRESS specification:

```
*)
ENTITY applied_security_classification_assignment
  SUBTYPE OF (security_classification_assignment);
  items : SET [1:?] OF security_classification_item;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the **security_classification** is assigned.

5.2.3.1.23 area_presentation_relationship

An **area_presentation_relationship** is a **representation_relationship** that relates two view areas. See ARM definition for View_area_relationship in paragraph 4.2.493 for more information.

EXPRESS specification:

```
*)
ENTITY area_presentation_relationship
  SUBTYPE OF (representation_relationship);
WHERE
  WR1:  ('AUTOMOTIVE_DESIGN.PRESENTATION_AREA' IN
         TYPEOF(SELF\representation_relationship.rep_1))
           AND
        ('AUTOMOTIVE_DESIGN.PRESENTATION_AREA' IN
         TYPEOF(SELF\representation_relationship.rep_2));
END_ENTITY;
(*
```

Formal propositions:

WR1: Both **representation**s related with an **area_presentation_relationship** shall be of type **presentation_area**.

5.2.3.1.24 auto_design_drawing_revision

An **auto_design_drawing_revision** is a type of **drawing_revision** that specifies constraints within the context of draughting.

EXPRESS specification:

```
*)
ENTITY auto_design_drawing_revision
  SUBTYPE OF (drawing_revision);
WHERE
  WR1:  -- drawing_requires_sheets:
        (SIZEOF (USEDIN (SELF,
           'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET')) >= 1)
        AND
        (SIZEOF (QUERY (ais <* USEDIN (SELF,
                 'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') |
          NOT ('AUTOMOTIVE_DESIGN.DRAWING_SHEET_REVISION'
                  IN TYPEOF (ais.area))
        )) = 0);
```

```
WR2:  -- drawing_title:
      SIZEOF (USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
                            'DRAUGHTING_TITLE.ITEMS')) <= 1;
WR3:  -- sheet_title:
      SIZEOF (QUERY (ais <* USEDIN (SELF,
              'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') |
        NOT (SIZEOF (USEDIN (ais.area,
                            'AUTOMOTIVE_DESIGN.' +
                            'DRAUGHTING_TITLE.ITEMS')) <= 1)
      )) = 0;
WR4:  -- allowable_items_of_sheet:
      SIZEOF (QUERY (ais <* USEDIN (SELF,
              'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') |
        NOT (SIZEOF (QUERY (item <* ais.area.items |
          NOT (SIZEOF (TYPEOF(item) *
               ['AUTOMOTIVE_DESIGN.STYLED_ITEM',
                'AUTOMOTIVE_DESIGN.MAPPED_ITEM',
                'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT',
                'AUTOMOTIVE_DESIGN.PLANAR_BOX']) = 1)
            )) = 0)
      )) = 0;
WR5:  -- sheet_contains_view_or_annotation:
      SIZEOF (QUERY (ais <* USEDIN (SELF,
              'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') |
        NOT (SIZEOF (QUERY (item <* ais.area.items |
              (SIZEOF (TYPEOF(item) *
               ['AUTOMOTIVE_DESIGN.STYLED_ITEM',
                'AUTOMOTIVE_DESIGN.MAPPED_ITEM']) = 1)
             )) > 0)
      )) = 0;
WR6:  -- planar_box_in_sheet:
      SIZEOF (QUERY (ais <* USEDIN (SELF,
              'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') |
        NOT (SIZEOF (QUERY (p_b <*
          QUERY (item <* ais.area.items |
             ('AUTOMOTIVE_DESIGN.PLANAR_BOX'
                 IN TYPEOF(item))) |
             NOT (SIZEOF (USEDIN (p_b, 'AUTOMOTIVE_DESIGN.PRESENTATION_SIZE.SIZE')
             ) = 1)
        )) = 0)
      )) = 0;
WR7:  -- sheets_contain_views:
      SIZEOF (QUERY (ais <* USEDIN (SELF,
              'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') |
        NOT (SIZEOF (QUERY (mi <* QUERY (item <* ais.area.items |
               ('AUTOMOTIVE_DESIGN.MAPPED_ITEM'
                       IN TYPEOF(item))) |
          NOT ('AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN
                  TYPEOF(mi.mapping_source.mapped_representation))
        )) = 0)
      )) = 0;
WR8:  -- sheets_contain_placements:
```

```
      SIZEOF (QUERY (ais <* USEDIN (SELF,
              'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') |
        NOT (SIZEOF (QUERY (a2p <* QUERY (item <*
                                          ais.area.items |
               ('AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT'
                    IN TYPEOF(item))) |
          NOT (SIZEOF (USEDIN (a2p,
                                 'AUTOMOTIVE_DESIGN.' +
                                 'MAPPED_ITEM.MAPPING_TARGET')) > 0)
        )) = 0)
      )) = 0;
WR9:  -- sheets_have_unique_context:
      SIZEOF (QUERY (ais <* USEDIN (SELF,
              'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') |
        NOT (SIZEOF (ais.area.context_of_items.
                     representations_in_context) = 1)
      )) = 0;
WR10: -- views_presented_once:
      SIZEOF (QUERY (ais <* USEDIN (SELF,
             'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') |
        NOT (SIZEOF (QUERY (mi <* QUERY (item <* ais.area.items |
               ('AUTOMOTIVE_DESIGN.MAPPED_ITEM'
                   IN TYPEOF(item))) |
          NOT (SIZEOF (USEDIN (mi.mapping_source.mapped_representation,
                           'AUTOMOTIVE_DESIGN.' +
                           'REPRESENTATION_MAP.MAPPED_REPRESENTATION')) = 1)
        )) = 0)
      )) = 0;
WR11: -- allowable_items_of_view:
      SIZEOF (QUERY (ais <* USEDIN (SELF,
              'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') |
        NOT (SIZEOF (QUERY (mi <* QUERY (item <* ais.area.items |
               ('AUTOMOTIVE_DESIGN.MAPPED_ITEM'
                   IN TYPEOF(item))) |
          NOT (SIZEOF (QUERY (pv_item <* mi.mapping_source.
                              mapped_representation.items |
            NOT (SIZEOF (TYPEOF(pv_item) *
                      ['AUTOMOTIVE_DESIGN.' +
                       'STYLED_ITEM',
                       'AUTOMOTIVE_DESIGN.CAMERA_IMAGE',
                       'AUTOMOTIVE_DESIGN.' +
                       'AXIS2_PLACEMENT']) = 1)
          )) = 0)
        )) = 0)
      )) = 0;
WR12: -- views_contain_one_projection:
      SIZEOF (QUERY (ais <* USEDIN (SELF,
              'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') |
        NOT (SIZEOF (QUERY (mi <* QUERY (item <* ais.area.items |
               ('AUTOMOTIVE_DESIGN.MAPPED_ITEM'
                    IN TYPEOF(item))) |
          NOT (SIZEOF (QUERY (pv_item <* mi.mapping_source.
```

```
                                mapped_representation.items |
                ('AUTOMOTIVE_DESIGN.CAMERA_IMAGE'
                    IN TYPEOF(pv_item))
           )) = 1)
         )) = 0)
       )) = 0;
  WR13: -- views_contain_one_placement:
       SIZEOF (QUERY (ais <* USEDIN (SELF,
               'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') |
         NOT (SIZEOF (QUERY (mi <* QUERY (item <* ais.area.items |
                 ('AUTOMOTIVE_DESIGN.MAPPED_ITEM'
                     IN TYPEOF(item))) |
           NOT (SIZEOF (QUERY (a2p <* QUERY (pv_item <* mi.mapping_source.
                                mapped_representation.items |
                 ('AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT'
                       IN TYPEOF(pv_item))) |
             ('AUTOMOTIVE_DESIGN.' +
              'MAPPING_SOURCE.MAPPING_ORIGIN' IN ROLESOF (a2p))
           )) = 1)
         )) = 0)
       )) = 0;
  WR14: -- views_have_unique_context:
       SIZEOF (QUERY (ais <* USEDIN (SELF,
               'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') |
         NOT (SIZEOF (QUERY (mi <* QUERY (item <* ais.area.items |
                 ('AUTOMOTIVE_DESIGN.MAPPED_ITEM'
                     IN TYPEOF(item))) |
           NOT (SIZEOF (mi.mapping_source.
                           mapped_representation.context_of_items.
                           representations_in_context) = 1)
         )) = 0)
       )) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **auto_design_drawing_revision** shall be related to at least one **presentation_area**, each of which is a **drawing_sheet_revision.**

WR2: The **auto_design_drawing_revision** shall be one of the items of no more than one **draughting_title.**

WR3: Each sheet of the **auto_design_drawing_revision** shall be one of the items of no more than one **draughting_title.**

WR4: The items of each sheet of the **auto_design_drawing_revision** shall be a **styled_item**, a **mapped_item**, an **axis2_placement**, or a **planar_box.**

WR5: The items of each sheet of the **auto_design_drawing_revision** shall include at least one **styled_item** or **mapped_item.**

WR6: Each sheet of the **auto_design_drawing_revision** shall contain exactly one **planar_box** that is the size of a **presentation_size.**

WR7: If there is a **mapped_item** in a sheet of the **auto_design_drawing_revision**, the source of the **mapped_item** shall be a **presentation_view.**

WR8: Every **axis2_placement** in a sheet of the **auto_design_drawing_revision** shall be the **mapping_target** for some **mapped_item.**

WR9: The **representation_context** of a sheet of the **auto_design_drawing_revision** shall not be the context of any other representation.

WR10: Each view in a sheet of the **auto_design_drawing_revision** shall be the **mapped_representation** for exactly one **mapped_item.**

WR11: The items of each view of each sheet of the **auto_design_drawing_revision** shall be a **styled_item**, a **camera_image**, or an **axis2_placement.**

WR12: Each view of each sheet of the **auto_design_drawing_revision** shall contain exactly one **camera_image.**

WR13: Each view of each sheet of the **auto_design_drawing_revision** shall contain exactly one **axis2_placement** which is the **mapping_origin** for a **mapped_item.**

WR14: The **representation_context** of each view of each sheet of the **auto_design_drawing_revision** shall not be the context of any other representation.

5.2.3.1.25 bead

A **bead** is a type of **feature_definition** applicable to panel parts that is used to increase the stiffness of the part. See ARM definition for Bead in paragraph 4.2.39 for more information.

EXPRESS specification:

```
*)
ENTITY bead
  SUBTYPE OF (feature_definition);
WHERE
  WR1:   SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION.DEFINITION') |
                 SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                          TYPEOF(pdr.used_representation)) AND
                         ({1 <= SIZEOF(pdr.used_representation) <= 3})
                       )) = 1
         )) = 1;
  WR2:   SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION.DEFINITION') |
                 SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
```

```
                        TYPEOF(pdr.used_representation)) AND
                       (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
                               srwp_i.name IN ['orientation','maximum depth',
                                               'base radius']
                        )) = SIZEOF(pdr.used_representation.items) )
               )) = 1
      )) = 1;
WR3:   SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                          'PROPERTY_DEFINITION.DEFINITION') |
           SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                 TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
                         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                  'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
                                  TYPEOF(srwp_i)
                          ) = 2 ) AND
                         (srwp_i.name = 'maximum depth')
                 )) <= 1 )
           )) = 1
      )) = 1;
WR4:   SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                          'PROPERTY_DEFINITION.DEFINITION') |
           SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                 TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
                         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                  'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
                                  TYPEOF(srwp_i)
                          ) = 2 ) AND
                         (srwp_i.name = 'base radius')
                 )) <= 1 )
           )) = 1
      )) = 1;
WR5:   SIZEOF( QUERY( sa <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                          'SHAPE_ASPECT.OF_SHAPE') |
               (sa.name = 'sweep path') AND
               (sa.description = 'path feature component occurrence') AND
               (SIZEOF( QUERY(sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
                                  'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
                        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
                        TYPEOF(sar)
                      )) = 1 ) AND
               (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
                                 'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
                        ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
                         TYPEOF(sar)) AND
                        (sar.name = 'path feature component usage') AND
                        ('AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN
```

```
                        TYPEOF(sar.relating_shape_aspect)) AND
                       (sar.relating_shape_aspect.description = 'linear')
                )) = 1 )
        )) = 1;
WR6:    SIZEOF( QUERY( sa <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'SHAPE_ASPECT.OF_SHAPE') |
                (sa.name = 'bead profile') AND
                (sa.description = 'open profile occurrence') AND
                (SIZEOF( QUERY(sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
                                      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
                         'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
                         TYPEOF(sar)
                       )) = 1 ) AND
                (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
                                'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
                         ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
                          TYPEOF(sar)) AND
                         (sar.name = 'open profile usage') AND
                         (SIZEOF(['AUTOMOTIVE_DESIGN.OPEN_PATH_PROFILE',
                                  'AUTOMOTIVE_DESIGN.PARTIAL_CIRCULAR_PROFILE',
                                  'AUTOMOTIVE_DESIGN.ROUNDED_U_PROFILE',
                                  'AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE',
                                  'AUTOMOTIVE_DESIGN.TEE_PROFILE',
                                  'AUTOMOTIVE_DESIGN.VEE_PROFILE'] *
                                  TYPEOF(sar.relating_shape_aspect))
                          = 1 )
                )) = 1 )
        )) = 1;
WR7:    SIZEOF( QUERY( sa <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'SHAPE_ASPECT.OF_SHAPE') |
                (sa.name = 'end condition at path start') AND
                (sa.description = 'bead end occurrence') AND
                (SIZEOF( QUERY(sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
                                      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
                         'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
                         TYPEOF(sar)
                       )) = 1 ) AND
                (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
                                'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
                         ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
                         TYPEOF(sar)) AND
                         (sar.name = 'bead end usage') AND
                         ('AUTOMOTIVE_DESIGN.BEAD_END' IN
                          TYPEOF(sar.relating_shape_aspect))
                ))  = 1 )
        )) = 1;
WR8:    SIZEOF( QUERY( sa <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'SHAPE_ASPECT.OF_SHAPE') |
                (sa.name = 'end condition at path end') AND
                (sa.description = 'bead end occurrence') AND
                (SIZEOF( QUERY(sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
                                      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
```

```
                        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
                        TYPEOF(sar)
                     )) = 1 ) AND
               (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
                                 'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
                        ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
                        TYPEOF(sar)) AND
                        (sar.name = 'bead end usage') AND
                        ('AUTOMOTIVE_DESIGN.BEAD_END' IN
                         TYPEOF(sar.relating_shape_aspect))
                )) = 1 )
        )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The implicit representation of the **bead** shall contain between one and three **representation_items** in its set of **items**.

WR2: The implicit representation of a **bead** shall only contain **representation_items** in its set of **items** that have a **name** of either 'orientation', 'maximum depth', or 'base radius'.

WR3: The implicit representation of the **bead** shall contain at most one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'maximum depth'.

WR4: The implicit representation of the **bead** shall contain at most one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'base radius'.

WR5: The **bead** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'sweep path' and a **description** of 'path feature component occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'path feature component usage' in which the **relating_shape_aspect** is a **path_feature_component** with a **description** of 'linear'.

WR6: The **bead** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'bead profile' and a **description** of 'open profile occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'open profile usage' in which the **relating_shape_aspect** is an **open_path_profile**, **partial_circular_profile**, **rounded_u_profile**, **square_u_profile**, **tee_profile**, or **vee_profile.**

WR7: The **bead** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'end condition at path start' and a **description** of 'bead end occurrence' that is the **related_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'bead end usage' in which the **relating_shape_aspect** is a **bead_end.**

WR8: The **bead** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'end condition at path end' and a **description** of 'bead end occurrence' that is the **related_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'bead end usage' in

which the **relating_shape_aspect** is a **bead_end.**

5.2.3.1.26 bead_end

A **bead_end** is a type of **shape_aspect** that is the representation of the end condition for a **bead_feature.** See ARM definition for Bead_end_type in paragraph 4.2.38, for Open_bead_end_type in paragraph 4.2.291, for Radiused_bead_end_type in paragraph 4.2.366, and for Linear_bead_end_type in paragraph 4.2.266 for more information.

EXPRESS specification:

```
*)
ENTITY bead_end
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
        TYPEOF(SELF.of_shape.definition);
  WR2:   SELF.description IN ['linear','open','radiused'];
  WR3:  (SELF.description <> 'linear') XOR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                 'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                 TYPEOF(pdr.used_representation)
                )) > 0 );
  WR4:  (SELF.description <> 'linear') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                  ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                   TYPEOF(pdr.used_representation)) AND
                  (SIZEOF(pdr.used_representation.items) = 2)
                )) = 1 );
  WR5:  (SELF.description <> 'linear') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                 ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                  TYPEOF(pdr.used_representation)) AND
                 (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                          (srwp_i.name = 'end length') AND
                          (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                   'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
                                   TYPEOF(srwp_i)
                                 ) = 2 )
                        )) = 1 )
                )) = 1 );
  WR6:  (SELF.description <> 'linear') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                 ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                  TYPEOF(pdr.used_representation)) AND
                 (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
```

```
                        (srwp_i.name = 'end shape') AND
                        (srwp_i.description IN ['straight','tapered']) AND
                        ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                         TYPEOF(srwp_i))
                      )) = 1 )
              )) = 1 );
  WR7:   SIZEOF( QUERY( fcr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
                ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
                 TYPEOF(fcr)) AND
                (fcr.name = 'bead end usage') AND
                ('AUTOMOTIVE_DESIGN.BEAD' IN
                 TYPEOF(fcr.related_shape_aspect.of_shape.definition)) AND
                (fcr.related_shape_aspect.of_shape.definition.name IN
                 ['end condition at path start','end condition at path end']) AND
                (fcr.related_shape_aspect.of_shape.definition.description =
                 'bead end occurrence')
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **bead_end** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **description** for the **bead_end** shall be either 'linear', 'open', or 'radiused'.

WR3: A **bead_end** has an implicit representation if and only if it has a **description** of 'linear'.

WR4: If the **description** of the **bead_end** is 'linear', its implicit representation shall contain exactly two **representation_items** in its set of **items**.

WR5: If the **description** of the **bead_end** is 'linear', exactly one **representation_item** used for the implicit representation of the **bead_end** shall be of type **measure_representation_-item** and **length_measure_with_unit** with a name of 'end length'.

WR6: If the **description** of the **bead_end** is 'linear', exactly one **representation_item** used for the implicit representation of the **bead_end** shall be of type **descriptive_representation_-item** with a name of 'end shape' and a **description** of either 'straight' or 'tapered'.

WR7: The **bead_end** shall be the **relating_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'bead end usage' in which the **related_shape_aspect** is a **shape_aspect** of a **bead** with a **name** of either 'end condition at path start' or 'end condition at path end' and a **description** of 'bead end occurrence'.

5.2.3.1.27 boss

A **boss** is a type of **feature_definition** that is the representation of an enclosed shape protrusion that is added to a solid base shape. This added volume is represented by sweeping a closed profile along a linear path represented by a **path_feature_component.** The top of the **boss** is represented by a **boss_top.** The walls of the **boss** may have a slope that is represented by a

taper. See ARM definition for Boss_feature in paragraph 4.2.42, for Circular_boss in paragraph 4.2.56, for Rectangular_boss in paragraph 4.2.368, and for General_boss in paragraph 4.2.190 for more information.

EXPRESS specification:

```
*)
ENTITY boss
  SUBTYPE OF (feature_definition);
WHERE
  WR1:  SELF.description IN ['circular','rectangular','general'];
  WR2:  (SELF.description <> 'circular') OR
        (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pds)) AND
          (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
         'SHAPE_ASPECT.OF_SHAPE') |
           (sa.name = 'boundary') AND
           (sa.description = 'closed profile occurrence') AND
           (SIZEOF( QUERY( sdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT_RELATIONSHIP.' +
          'RELATED_SHAPE_ASPECT') |
            ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
              TYPEOF(sdr)) AND
            (sdr.name = 'closed profile occurrence') AND
            ('AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE' IN
             TYPEOF(sdr.relating_shape_aspect))
          )) = 1 )
         )) = 1 )
         )) = 1);
  WR3:  (SELF.description <> 'general') OR
        (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pds)) AND
          (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
             'SHAPE_ASPECT.OF_SHAPE') |
           (sa.name = 'boundary') AND
           (sa.description = 'closed profile occurrence') AND
           (SIZEOF( QUERY( sdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT_RELATIONSHIP.' +
          'RELATED_SHAPE_ASPECT') |
            ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
              TYPEOF(sdr)) AND
            (sdr.name = 'closed profile usage') AND
            ('AUTOMOTIVE_DESIGN.CLOSED_PATH_PROFILE' IN
              TYPEOF(sdr.relating_shape_aspect))
          )) = 1 )
         )) = 1 )
         )) = 1);
```

```
WR4:  (SELF.description <> 'rectangular') OR
      (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
         TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
           'SHAPE_ASPECT.OF_SHAPE') |
         (sa.name = 'boundary') AND
         (sa.description = 'closed profile occurrence') AND
         (SIZEOF( QUERY( sdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
            TYPEOF(sdr)) AND
          (sdr.description = 'closed profile usage') AND
          ('AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE' IN
            TYPEOF(sdr.relating_shape_aspect))
        )) = 1 )
       )) = 1 )
       )) = 1);
WR5:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
       SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.' +
             'DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        ({1 <= SIZEOF(pdr.used_representation.items) <= 2} )
             )) = 1
            )) = 1;
WR6:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
       SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.' +
             'DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
         (srwp_i.name = 'orientation') OR
         (srwp_i.name = 'fillet radius')
              )) = SIZEOF(pdr.used_representation.items))
             )) = 1
            )) = 1;
WR7:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
       SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.' +
             'DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
         (srwp_i.name = 'fillet radius')
```

```
              )) <= 1 ) AND
        ((SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
          (srwp_i.name = 'fillet radius')
        )) = 0 ) OR
         (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
          (srwp_i.name = 'fillet radius') AND
          (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
           'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
          * TYPEOF(srwp_i)) = 2 )
        )) = 1 ))
              )) = 1
            )) = 1;
WR8:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
           'SHAPE_ASPECT.OF_SHAPE') |
         (sa.name = 'sweep path') AND
         (sa.description = 'path feature component occurrence') AND
         (SIZEOF( QUERY( sdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
            TYPEOF(sdr)) AND
          (sdr.name = 'path feature component occurrence') AND
          ('AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN
           TYPEOF(sdr.relating_shape_aspect)) AND
          (sdr.relating_shape_aspect.description =
           'linear')
        )) = 1 )
              )) = 1 )
            )) = 1;
WR9:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
           'SHAPE_ASPECT.OF_SHAPE') |
         (sa.name = 'top condition') AND
         (sa.description = 'boss top occurrence') AND
         (SIZEOF( QUERY( sdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(sdr)) AND
          (sdr.name = 'boss top usage') AND
          ('AUTOMOTIVE_DESIGN.BOSS_TOP' IN
           TYPEOF(sdr.relating_shape_aspect))
         )) = 1 )
              )) = 1 )
            )) = 1;
```

```
  WR10: SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pds)) AND
         (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
            'SHAPE_ASPECT.OF_SHAPE') |
          (sa.name = 'taper') AND
          (sa.description = 'taper occurrence') AND
          (SIZEOF( QUERY( sdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
         'SHAPE_ASPECT_RELATIONSHIP.' +
         'RELATED_SHAPE_ASPECT') |
           ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
             TYPEOF(sdr)) AND
           (sdr.name = 'taper usage') AND
           ('AUTOMOTIVE_DESIGN.TAPER' IN
            TYPEOF(sdr.relating_shape_aspect))
         )) = 1 )
                )) <= 1 )
              )) <= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **description** for the **boss** shall be either 'circular', 'rectangular', or 'general'.

WR2: If the **boss** has a **description** of 'circular' the **boss** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'boundary' and a **description** of 'closed profile occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'closed profile usage' in which the **relating_shape_aspect** is a **circular_closed_profile.**

WR3: If the **boss** has a **description** of 'general', the **boss** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'boundary' and a **description** of 'closed profile occurrence' that is the **related_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'closed profile usage' in which the **relating_shape_aspect** is a **closed_path_profile.**

WR4: If the **boss** has a **description** of 'rectangular' the **boss** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'boundary' and a **description** of 'closed profile occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'closed profile usage' in which the **relating_shape_aspect** is a **rectangular_closed_profile.**

WR5: The implicit representation of the **boss** shall contain between one and two **representation_items.**

WR6: The implicit representation of the **boss** shall contain only **representation_items** that have a **name** of either 'orientation' or 'fillet radius'.

WR7: At most one **representation_item** used for the implicit representation of the **boss** shall have a **name** of 'fillet radius'. If present, this **representation_item** shall be of type **measure_representation_item** and **length_measure_with_unit.**

WR8: The **boss** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'sweep path' and a **description** of 'path feature component occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'path feature component usage' in which the **relating_shape_aspect** is a **path_feature_component** with a **description** of 'linear'.

WR9: The **boss** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'top condition' and a **description** of 'boss top occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'boss top usage' in which the **relating_shape_aspect** is a **boss_top.**

WR10: The **boss** shall be the basis shape for at most one **shape_aspect** with a **name** of 'taper' and a **description** of 'taper occurrence'. If present, this **shape_aspect** shall be the **related_shape_aspect** in exactly one **feature_component_relationship** in with a **name** of 'taper usage' in which the **relating_shape_aspect** is a **taper**.

5.2.3.1.28 boss_top

A **boss_top** is a type of **shape_aspect** that is the end condition for a **boss**. See ARM definition for Boss_top_condition in paragraph 4.2.43, for Planar_top_condition in paragraph 4.2.316, and for General_top_condition in paragraph 4.2.209 for more information.

EXPRESS specification:

```
*)
ENTITY boss_top
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
        TYPEOF(SELF.of_shape.definition);
  WR2:  SELF.description IN ['planar','non-planar'];
  WR3:  (SELF.description <> 'planar') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                  'AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN
                  TYPEOF(pdr.used_representation)
                )) = 1 );
  WR4:  (SELF.description <> 'planar') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                  'AUTOMOTIVE_DESIGN.LOCATION_SHAPE_REPRESENTATION' IN
                  TYPEOF(pdr.used_representation)
                )) = 1 );
  WR5:  (SELF.description <> 'non-planar') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                  ('AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION' IN
                   TYPEOF(pdr.used_representation)) OR
                  ('AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION' IN
                   TYPEOF(pdr.used_representation))
```

```
              )) = 1 );
  WR6:  SIZEOF( QUERY( fcr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
                 ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
                   TYPEOF(fcr)) AND
                 (fcr.name = 'boss top usage') AND
                 ('AUTOMOTIVE_DESIGN.BOSS' IN
                  TYPEOF(fcr.related_shape_aspect.of_shape.definition)) AND
                 (fcr.related_shape_aspect.name = 'top condition') AND
                 (fcr.related_shape_aspect.description = 'boss top occurrence')
              )) = 0;
  WR7:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                TYPEOF(pdr.used_representation)
              )) = 1 ;
  WR8:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                 TYPEOF(pdr.used_representation)) AND
                (SIZEOF(pdr.used_representation.items) = 1)
              )) = 1 ;
  WR9:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                 TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                          TYPEOF(srwp_i)) AND
                         (srwp_i.name = 'top location') AND
                         (srwp_i.description IN ['part start','part end'])
                       )) = 1 )
              )) = 1 ;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **boss_top** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **description** of the **boss_top** shall be either 'planar' or 'non-planar'.

WR3: If the **description** of the **boss_top** is 'planar', the **boss_top** shall have exactly one **direction_shape_representation.**

WR4: If the **description** of the **boss_top** is 'planar', the **boss_top** shall have exactly one **location_shape_representation.**

WR5: If the **description** of the **boss_top** is 'non-planar', the **boss_top** shall either have exactly one **face_shape_representation** or exactly one **connected_face_shape_representation.**

WR6: The **boss_top** shall be the **relating_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'boss top usage' in which the **related_shape_aspect** is a **shape_aspect** of a **boss** with a **name** of 'top condition' and a **description** of 'boss top occurrence'.

WR7: The **boss_top** shall have exactly one **shape_representation_with_parameters** to specify its implicit representation.

WR8: The **shape_representation_with_parameters** that represents the **boss_top** shall contain exactly one **representation_item** in its set of **items**.

WR9: Exactly one **representation_item** used for the implicit representation of a **boss_top** shall be of type **descriptive_representation_item** with a **name** of 'top location' and a **description** of either 'path start' or 'path end'.

5.2.3.1.29 camera_image_2d_with_scale

A **camera_image_2d_with_scale** is a type of **camera_image** with a derived scale. The scale is the ratio between the size of the viewport and the size of the view window of the **camera_model.**

NOTE 1 – The viewport specifies the rectangular boundary of the view as it is depicted in the drawing sheet.

NOTE 2 – The definition of this entity is identical to the definition in ISO 10303-202.

EXPRESS specification:

```
*)
ENTITY camera_image_2d_with_scale
  SUBTYPE OF (camera_image);
DERIVE
  scale: positive_ratio_measure :=
          ((SELF\mapped_item.mapping_target\planar_extent.size_in_x) /
           (SELF\mapped_item.mapping_source.
            mapping_origin\camera_model_d2.view_window.size_in_x));
WHERE
  WR1:  ('AUTOMOTIVE_DESIGN.CAMERA_MODEL_D2' IN
          TYPEOF(SELF\mapped_item.mapping_source.mapping_origin));
  WR2:  aspect_ratio (SELF\mapped_item.mapping_target) =
          aspect_ratio (SELF\mapped_item.mapping_source.
                              mapping_origin\camera_model_d2.view_window);
  WR3:  SELF\mapped_item.mapping_source.
              mapping_origin\camera_model_d2.view_window_clipping;
END_ENTITY;
(*
```

Attribute definitions:

scale: The **positive_ratio_measure** derived from the rectangular size of the viewport and the rectangular size of the **view_window** of the **camera_model_d2.**

Formal propositions:

WR1: The source of the projection shall be a **camera_model_d2.**

WR2: The aspect ratio of the viewport shall equal the aspect ratio of the **view_window** of the **camera_model.**

WR3: The **view_window_clipping** attribute of the **camera_model_d2** shall be true.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **camera_image_2d_-with_scale** entity:

– restrict_multi_language_for_camera_image_2d_with_scale (see 5.2.4.90).

5.2.3.1.30 chamfer

A **chamfer** is a type of **transition_feature** that is the representation of a linear transition between two joining non-coplanar faces. See ARM definition for Chamfer in paragraph 4.2.50 for more information.

EXPRESS specification:

```
*)
ENTITY chamfer
  SUBTYPE OF (transition_feature);
WHERE
  WR1:  SIZEOF( QUERY (fcr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
            ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
              TYPEOF(fcr)) AND
            (fcr.name = 'first chamfer offset') AND
            (fcr.related_shape_aspect.description = 'chamfer offset') AND
            ('AUTOMOTIVE_DESIGN.CHAMFER_OFFSET' IN
              TYPEOF (fcr.related_shape_aspect))
         )) = 1;
  WR2:  SIZEOF( QUERY (fcr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(fcr)) AND
          (fcr.name = 'second chamfer offset') AND
          (fcr.related_shape_aspect.description IN
            ['chamfer offset','chamfer angle']) AND
          ('AUTOMOTIVE_DESIGN.CHAMFER_OFFSET' IN
            TYPEOF (fcr.related_shape_aspect))
               )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **chamfer** shall be the **relating_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'first chamfer offset' in which the **related_shape_aspect** is a **chamfer_offset** with a **description** of 'chamfer offset'.

WR2: The **chamfer** shall be the **relating_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'second chamfer offset' in which the **related_shape_aspect** is a **chamfer_offset** with a **description** of either 'chamfer offset' or 'chamfer angle'.

5.2.3.1.31 chamfer_offset

A **chamfer_offset** is a type of **shape_aspect** that is the representation of the linear setbacks from the edge of two surfaces to create a chamfer feature. See ARM definition for Chamfer_offset in paragraph 4.2.53 and for Chamfer_angle in paragraph 4.2.51 for more information.

EXPRESS specification:

```
*)
ENTITY chamfer_offset
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  SELF.description IN ['chamfer offset','chamfer angle'];
  WR2:  SIZEOF(QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF (pdr.used_representation)
         )) = 1;
  WR3:  SIZEOF( QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF (pdr.used_representation)) AND
           (SIZEOF(pdr.used_representation.items) = 1)
         )) = 1;
  WR4:  (SELF.description <> 'chamfer offset') OR
        (SIZEOF( QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF (pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
           (srwp_i.name = 'offset amount') AND
           (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
            'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 )
         )) = 1 )
              )) = 1 );
  WR5:  (SELF.description = 'chamfer angle') OR
        (SIZEOF( QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
```

```
          (srwp_i.name = 'offset angle') AND
          (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
           'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
          * TYPEOF(srwp_i)) = 2 )
        )) = 1 )
              )) = 1 );
  WR6:        SIZEOF(QUERY( sar <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
              'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
              TYPEOF(sar)
            )) > 0;
  WR7:  (SELF.description = 'chamfer offset') OR
        (SIZEOF( QUERY( fcr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(fcr)) AND
          (fcr.name <> 'first chamfer offset') AND
          (fcr.name <> 'second chamfer offset')
              )) = 0 );
  WR8:  (SELF.description = 'chamfer angle') OR
        (SIZEOF( QUERY( fcr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(fcr)) AND
          (fcr.name <> 'second chamfer offset')
              )) = 0 );
  WR9:  SIZEOF( QUERY( fcr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
               'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
         ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
           TYPEOF(fcr)) AND
         ('AUTOMOTIVE_DESIGN.CHAMFER' IN
           TYPEOF(fcr.relating_shape_aspect)) OR
         (('AUTOMOTIVE_DESIGN.HOLE_BOTTOM' IN
            TYPEOF(fcr.relating_shape_aspect)) AND
          (fcr.relating_shape_aspect.description = 'flat with chamfer')) OR
         (('AUTOMOTIVE_DESIGN.SHAPE_ASPECT' IN
            TYPEOF(fcr.relating_shape_aspect)) AND
          (fcr.relating_shape_aspect.description = 'top chamfer') AND
          ('AUTOMOTIVE_DESIGN.ROUND_HOLE' IN
            TYPEOF(fcr.relating_shape_aspect.of_shape.definition)))
         )) =
        SIZEOF( QUERY( sar <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
         'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
         'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
          TYPEOF(sar) ));
END_ENTITY;
(*
```

Formal propositions:

WR1: The **description** for the **chamfer_offset** shall be either 'chamfer offset' or 'chamfer angle'.

WR2: The **chamfer_offset** shall have its implicit representation specified by exactly one **shape_representation_with_parameters.**

WR3: The **chamfer_offset** shall have an implicit representation that contains exactly one **representation_item** in its set of **items**.

WR4: If the **description** of the **chamfer_offset** is 'chamfer offset', exactly one **representation_item** used for the implicit representation of the **chamfer_offset** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'offset amount'.

WR5: If the **description** of the **chamfer_offset** is 'chamfer angle', exactly one **representation_item** used for the implicit representation of the **chamfer_offset** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'offset angle'.

WR6: The **chamfer_offset** shall be the **related_shape_aspect** in at least one **feature_component_relationship.**

WR7: If the **description** of the **chamfer_offset** is 'chamfer offset', it shall be the **related_shape_aspect** only in **feature_component_relationship** instances with a **name** of 'first chamfer offset' or 'second chamfer offset'.

WR8: If the **description** of the **chamfer_offset** is 'chamfer angle', it shall be the **related_shape_aspect** only in **feature_component_relationship** instances with a **name** of 'second chamfer offset'.

WR9: The **chamfer_offset** shall be the **related_shape_aspect** only in **feature_component_relationship** instances in which the **relating_shape_aspect** is either a **chamfer**, a **hole_bottom** with a **description** of 'flat with chamfer', or a **shape_aspect** with a **description** of 'top chamfer' that is the **shape_aspect** of the shape of a **round_hole.**

5.2.3.1.32 characterized_product_category

A **characterized_product_category** is a type of **product_category** and of **characterized_object** that characterizes products of the same kind independent from their usage. See ARM definition for General_classification in paragraph 4.2.192 for more information.

EXPRESS specification:

```
*)
ENTITY characterized_product_category
  SUBTYPE OF (characterized_object, product_category);
WHERE
  WR1:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
         (pd.description <> 'category attribute') OR
```

```
        (SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         (pdr.used_representation.description <>
           'classification attribute') AND
         NOT ('AUTOMOTIVE_DESIGN.' +
             'EXTERNALLY_DEFINED_CLASSIFICATION_ATTRIBUTE' IN
             TYPEOF(pdr.used_representation))
               )) > 0 )
             )) = 0;
  WR2:  SIZEOF( QUERY( adr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'APPLIED_DOCUMENT_REFERENCE.ITEMS') |
        adr.role.name = 'classification source'
             )) <= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **characterized_product_category** shall be referenced as the **definition** only by instances of **property_definition** that have a **description** of 'category attribute' and that are referenced as the **definition** only by instances of **property_definition_representation** that reference as their **used_representation** a **representation** that either has a **description** of 'classification attribute' or is an **externally_defined_classification_attribute.**

WR2: The **characterized_product_category** shall be contained in the set of **items** of at most one **applied_document_reference** that references as its **role** a **document_role** that has a **name** of 'classification source'.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **characterized_product_category** entity:

– restrict_multi_language_for_characterized_product_category (see 5.2.4.93).

5.2.3.1.33 circular_closed_profile

A **circular_closed_profile** is a type of **shape_aspect** that consists of an enclosed 2D area and a reference system for the parameters of the **circular_closed_profile.** The enclosed area is circular and defined by a radius. The reference system is used to define location and orientation of the **circular_closed_profile** and consists of two orthogonal axes X and Y that intersect in a reference point. See ARM definition for Circular_closed_profile in paragraph 4.2.57 for more information.

EXPRESS specification:

```
*)
ENTITY circular_closed_profile
  SUBTYPE OF (shape_aspect);
WHERE
```

```
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF (SELF.of_shape.definition);
  WR2:  SIZEOF (QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
                                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                TYPEOF (pdr.used_representation)
              )) = 1;
  WR3:  SIZEOF (QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
                                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                SIZEOF(pdr.used_representation.items) = 2
              )) = 1;
  WR4:  SIZEOF (QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
                                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                       ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i)) AND
                       (srwp_i.name = 'orientation')
                     )) = 1
              )) = 1;
  WR5:  SIZEOF (QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
                                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                       (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                 'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
                                TYPEOF(srwp_i)
                               ) = 2 ) AND
                       (srwp_i.name = 'diameter')
                     )) = 1
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **circular_closed_profile** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **circular_closed_profile** shall have exactly one implicit representation defined by a **shape_representation_with_parameters.**

WR3: The **shape_representation_with_parameters** that represents the **circular_closed_profile** shall contain two **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **circular_closed_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: Exactly one **representation_item** used for the implicit representation of a **circular_closed_profile** shall be of type **measure_representation_item** and **length_angle_measure_with_unit** with a **name** of 'diameter'.

Informal propositions:

IP1: The **circular_closed_profile** shall be planar and shall lie in the X-Y plane of the reference system.

IP2: The reference point of the **circular_closed_profile** is defined at the centre of the circle.

5.2.3.1.34 circular_pattern

A **circular_pattern** is a type of **replicate_feature** that relates a base feature and one or more **shape_aspects** which are the placement of the base feature at a specified circular location on the base part. A diameter is defined with a placement on the part and a number of base features are equally spaced around this diameter. See ARM definition for Circular_pattern in paragraph 4.2.61 for more information.

EXPRESS specification:

```
*)
ENTITY circular_pattern
  SUBTYPE OF (replicate_feature);
WHERE
  WR1:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                    'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF (pdr.used_representation)) AND
         ({4 <= SIZEOF(pdr.use_representation.items) <= 5})
                     )) = 1
         )) = 1;
  WR2:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF (pdr.used_representation)) AND
                       (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                 (r_i.name = 'diameter') AND
                              (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
           'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
          * TYPEOF(r_i)  ) = 2 )
                )) = 1 )
               )) = 1
              )) <= 1;
  WR3:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF (pdr.used_representation)) AND
```

```
         (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
          (r_i.name = 'base feature rotation') AND
                        (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                 'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
                                 * TYPEOF(r_i) ) = 2 )
                )) = 1 )
               )) = 1
        )) = 1;
  WR4:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.' +
               'DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF (pdr.used_representation)) AND
         (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
          ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN
            TYPEOF(r_i)) AND
          ('AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN
            TYPEOF(r_i\measure_representation_item.value_component)) AND
          (r_i.name = 'number of features')
                                )) = 1 )
                     )) = 1
              )) = 1;
  WR5:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                                 'PROPERTY_DEFINITION.DEFINITION') |
                     SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.' +
               'DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                 TYPEOF (pdr.used_representation)) AND
                (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                         (r_i.name = 'angular spacing') AND
                         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                  'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
                                   * TYPEOF(r_i) ) = 2 )
                )) = 1 )
               )) = 1
        )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The implicit representation of the **circular_pattern** shall contain four or five **representation_items** in its set of **items**.

WR2: The implicit representation of the **circular_pattern** shall contain at most one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'diameter'.

WR3: The implicit representation of the **circular_pattern** shall contain exactly one **representation_item** of type **measure_representation_item** and **plane_angle_measure_with_unit** with a name of 'base feature rotation'.

WR4: The implicit representation of the **circular_pattern** shall contain exactly one **representation_item** of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'number of features'.

WR5: The implicit representation of the **circular_pattern** shall contain exactly one **representation_item** of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'angular spacing'.

Informal propositions:

IP1: The location of the **circular_pattern** shall be defined at the center of the circle.

IP2: The **circular_pattern** shall be defined in the X-Y plane, with the X direction intersecting the placement position of the first base feature.

5.2.3.1.35 classification_system

A **classification_system** is a type of group that describes a method for classifying objects. See ARM definition for Classification_system in paragraph 4.2.70 for more information.

EXPRESS specification:

```
*)
ENTITY classification_system
  SUBTYPE OF (group);
WHERE
  WR1:  (SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.APPLIED_DOCUMENT_REFERENCE.ITEMS')
               ) = 0 ) OR
        (SIZEOF( QUERY( adr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'APPLIED_DOCUMENT_REFERENCE.ITEMS') |
          adr.role.name = 'classification system source'
               )) = 1 );
END_ENTITY;
(*
```

Formal propositions:

WR1: The **classification_system** shall be contained in the set of **items** of at most one **applied_document_reference.** If present, this **applied_document_reference** shall reference as its **role** a **document_role** that has a **name** of 'classification system source'.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **classification_system** entity:

– restrict_multi_language_for_classification_system (see 5.2.4.94).

5.2.3.1.36 closed_path_profile

A **closed_path_profile** is a type of **shape_aspect** that consists of a connected set of curves and possibly a reference system that may be used to place the **closed_path_profile.** See ARM definition for General_closed_profile in paragraph 4.2.194 for more information.

EXPRESS specification:

```
*)
ENTITY closed_path_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SIZEOF (QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
                                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                TYPEOF (pdr.used_representation)
        )) <= 1;
  WR3:  SIZEOF (QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
                                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                 TYPEOF (pdr.used_representation)) AND
                (SIZEOF(pdr.used_representation.items) <> 1 )
        )) = 0;
  WR4:  SIZEOF (QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
                                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                 TYPEOF (pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.items |
                         ('AUTOMOTIVE_DESIGN.PLACEMENT' IN
                          TYPEOF(pdr.used_representation.items)) AND
                         (srwp_i.name = 'orientation')
                 )) = 1 )
        )) = 1;
  WR5:  SIZEOF (QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
                                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                'AUTOMOTIVE_DESIGN.PATH_SHAPE_REPRESENTATION' IN
                TYPEOF (pdr.used_representation)
        )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **closed_path_profile** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **closed_path_profile** shall have at most one implicit representation defined by a **shape_representation_with_parameters.**

WR3: The **shape_representation_with_parameters** that represents the **closed_path_profile** shall contain one **representation_item** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **closed_path_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: The **closed_path_profile** shall have exactly one **path_shape_representation.**

Informal propositions:

IP1: The **closed_path_profile** shall be planar and shall lie in the X-Y plane of the reference system if this is present.

5.2.3.1.37 component_function_definition

A **component_function_definition** is a type of **configuration_design** that specifies that a **product_component** fulfills a certain function. See ARM definition for Function_component_association in paragraph 4.2.189 for more information.

EXPRESS specification:

```
*)
ENTITY component_function_definition
  SUBTYPE OF (configuration_design);
WHERE
  WR1:  SELF.design.frame_of_reference.name = 'functional definition';
  WR2:  'AUTOMOTIVE_DESIGN.PRODUCT_COMPONENT' IN
         TYPEOF(SELF.configuration);
END_ENTITY;
(*
```

Formal propositions:

WR1: The **component_function_definition** shall reference as the **design** a **product_definition** that references as its **frame_of_reference** a **product_definition_context** that has a **name** of 'functional definition'.

WR2: The **component_function_definition** shall reference as the **configuration** a **configuration_item** that is of type **product_component.**

5.2.3.1.38 compound_feature_in_panel

A **compound_feature_in_panel** is a type of **feature_definition** that is the representation of several other features for the purpose of creating unique user defined features for application on

panel parts. See ARM definition for Compound_feature_in_panel in paragraph 4.2.81 for more information.

EXPRESS specification:

```
*)
ENTITY compound_feature_in_panel
  SUBTYPE OF (feature_definition);
WHERE
  WR1:  (SIZEOF(['AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
            'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT']
           * TYPEOF(SELF)) = 2 ) OR
        (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN-PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pds)) AND
          (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
                'SHAPE_ASPECT.OF_SHAPE') |
           'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN
            TYPEOF(sa)
         )) = 1 )
               )) = 1 );
  WR2:  (('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND
         (SIZEOF( QUERY( fcr <* SELF\composite_shape_aspect.component_relationships |
           'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(fcr) )) =
          SIZEOF(SELF\composite_shape_aspect.component_relationships))) OR
        (NOT ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND
         (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pds)) AND
           (SIZEOF( QUERY( csa <* USEDIN(pds, 'AUTOMOTIVE_DESIGN.' +
         'SHAPE_ASPECT.OF_SHAPE') |
            ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN
              TYPEOF(csa)) AND
            (SIZEOF( QUERY( fcr <* csa.component_relationships |
             'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
              TYPEOF(fcr) )) =
             SIZEOF(SELF\composite_shape_aspect.component_relationships))
          )) = 1 )
         )) = 1 ));
  WR3:  (SIZEOF(['AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_SOLID',
          'AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
          'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT']
         * TYPEOF(SELF)) <> 3 ) OR
        (SIZEOF( QUERY( sar <* SELF\composite_shape_aspect.component_relationships |
          SIZEOF(['AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
          'AUTOMOTIVE_DESIGN.TRANSITION_FEATURE']
         * TYPEOF(sar.related_shape_aspect)) = 1
               )) = SIZEOF(SELF\composite_shape_aspect.component_relationships));
```

```
WR4:  ('AUTOMOTIVE_DESIGN.INSTANCED_FEATURE' IN TYPEOF(SELF)) OR
      (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( csa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT.OF_SHAPE') |
         ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN
           TYPEOF(csa)) AND
         (SIZEOF( QUERY( sar <* csa.component_relationships |
          SIZEOF(['AUTOMOTIVE_DESIGN.PLACED_FEATURE',
          'AUTOMOTIVE_DESIGN.TRANSITION_FEATURE']
         * TYPEOF(sar.related_shape_aspect)) = 1
        )) <> SIZEOF(csa.component_relationships))
       )) > 0 )
             )) = 0 );
WR5:  (('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND
       (SIZEOF( QUERY( fcr <* SELF\composite_shape_aspect.component_relationships |
        ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
          TYPEOF(fcr.related_shape_aspect)) AND
        (SIZEOF( QUERY( pdr <* USEDIN( fcr.related_shape_aspect,
        'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
         (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
          ('AUTOMOTIVE_DESIGN.PLACEMENT' IN
            TYPEOF(srwp_i)) AND
          (srwp_i.name = 'element placement') )) = 1 )
       )) = 1 )
       )) = SIZEOF( QUERY( fcr <* SELF\composite_shape_aspect.component_relationships
      |
             ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
              TYPEOF(fcr.related_shape_aspect)) )) )) OR
      (NOT ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND
       (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
            ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
              TYPEOF(pds)) AND
            (SIZEOF( QUERY( csa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
       ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN
         TYPEOF(csa)) AND
       (SIZEOF( QUERY( fcr <* csa.component_relationships |
          ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
            TYPEOF(fcr.related_shape_aspect)) AND
          (SIZEOF( QUERY( pdr <* USEDIN( fcr.related_shape_aspect,
       'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
             ('AUTOMOTIVE_DESIGN.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
```

```
        TYPEOF(pdr.used_representation)) AND
              (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
               ('AUTOMOTIVE_DESIGN.PLACEMENT' IN
          TYPEOF(srwp_i)) AND
               (srwp_i.name = 'element placement') )) = 1 )
               )) = 1 )
        )) = SIZEOF( QUERY( fcr <* SELF\composite_shape_aspect.component_relationships |
           ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
             TYPEOF(fcr.related_shape_aspect)) )))
           )) = 1 )
        )) = 1 ));
END_ENTITY;
(*
```

Formal propositions:

WR1: Each instance of **compound_feature_in_panel** shall either be of type **instanced_feature** and **composite_shape_aspect**, or shall have a shape that has exactly one **shape_aspect** identified on it that is of type **composite_shape_aspect.**

WR2: The **shape_aspect_relationships** that are used to relate the single features with the **composite_shape_aspect** shall be **feature_component_relationships.**

WR3: If the **compound_feature_in_panel** is of type **instanced_feature** and **composite_shape_aspect** the **related_shape_aspects** defined by its **component_relationships** shall be of type **instanced_feature** or **transition_feature.**

WR4: If the **compound_feature_in_panel** is not of type **instanced_feature**, the **composite_shape_aspect** identified on its shape shall only have **related_shape_aspects** defined by its **component_relationships** that are of type **placed_feature** or **transition_feature.**

WR5: Each **placed_feature** that is specified by the **component_relationships** of a **composite_shape_aspect** of a **compound_feature_in_panel** shall have a **representation_item** of type **placement** with a **name** of 'element placement' in its **shape_representation_with_parameters.**

5.2.3.1.39 compound_feature_in_solid

A **compound_feature_in_solid** is a type of **feature_definition** that is the representation of several other features for the purpose of creating unique user defined features for application on solid parts. See ARM definition for Compound_feature_in_solid in paragraph 4.2.84 for more information.

EXPRESS specification:

```
*)
ENTITY compound_feature_in_solid
  SUBTYPE OF (feature_definition);
WHERE
```

```
WR1:  (SIZEOF(['AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_SOLID',
        'AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
        'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT']
       * TYPEOF(SELF)) = 3 ) OR
      (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pds)) AND
          (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
            'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN
            TYPEOF(sa)
         )) = 1 )
              )) = 1 );
WR2:  (('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND
       (SIZEOF( QUERY( fcr <* SELF\composite_shape_aspect.component_relationships |
         'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
         TYPEOF(fcr) )) =
        SIZEOF(SELF\composite_shape_aspect.component_relationships))) OR
      (NOT ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND
       (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
         TYPEOF(pds)) AND
        (SIZEOF( QUERY( csa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT.OF_SHAPE') |
       ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_SAPECT' IN
         TYPEOF(csa)) AND
       (SIZEOF( QUERY( fcr <* csa.component_relationships |
       'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(fcr) )) =
        SIZEOF(SELF\composite_shape_aspect.component_relationships))
              )) = 1 )
       )) = 1 ));
WR3:  (SIZEOF(['AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_SOLID',
        'AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
        'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT']
       * TYPEOF(SELF)) <> 3 ) OR
      (SIZEOF( QUERY( sar <* SELF\composite_shape_aspect.component_relationships |
        SIZEOF(['AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
        'AUTOMOTIVE_DESIGN.TRANSITION_FEATURE',
        'AUTOMOTIVE_DESIGN.THREAD_OCCURRENCE']
        * TYPEOF(sar.related_shape_aspect)) = 1
              )) = SIZEOF(SELF\composite_shape_aspect.component_relationships));
WR4:  ('AUTOMOTIVE_DESIGN.INSTANCED_FEATURE' IN TYPEOF(SELF)) OR
      (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
         TYPEOF(pds)) AND
        (SIZEOF( QUERY( csa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT.OF_SHAPE') |
        NOT ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN
```

```
              TYPEOF(csa)) OR
         (SIZEOF( QUERY( sar <* csa.component_relationships |
         SIZEOF(['AUTOMOTIVE_DESIGN.PLACED_FEATURE',
         'AUTOMOTIVE_DESIGN.TRANSITION_FEATURE',
         'AUTOMOTIVE_DESIGN.THREAD_OCCURRENCE']
        * TYPEOF(sar.related_shape_aspect)) = 1
              )) <> SIZEOF(csa.component_relationships))
              )) > 0 )
             )) = 0 );
WR5:  (('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND
       (SIZEOF( QUERY( fcr <* SELF\composite_shape_aspect.component_relationships |
        ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
          TYPEOF(fcr.related_shape_aspect)) AND
        (SIZEOF( QUERY( pdr <* USEDIN( fcr.related_shape_aspect,
        'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
         (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
          ('AUTOMOTIVE_DESIGN.PLACEMENT' IN
            TYPEOF(srwp_i)) AND
          (srwp_i.name = 'element placement') )) = 1 )
       )) = 1 )
       )) = SIZEOF( QUERY( fcr <* SELF\composite_shape_aspect.component_relationships |
            ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
             TYPEOF(fcr.related_shape_aspect)) )) )) OR
      (NOT ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND
       (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pds)) AND
           (SIZEOF( QUERY( csa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT.OF_SHAPE') |
       ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN
         TYPEOF(csa)) AND
       (SIZEOF( QUERY( fcr <* csa.component_relationships |
          ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
            TYPEOF(fcr.related_shape_aspect)) AND
          (SIZEOF( QUERY( pdr <* USEDIN( fcr.related_shape_aspect,
         'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              ('AUTOMOTIVE_DESIGN.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
              (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
       ('AUTOMOTIVE_DESIGN.PLACEMENT' IN
         TYPEOF(srwp_i)) AND
       (srwp_i.name = 'element placement') )) = 1 )
         )) = 1 )
       )) = SIZEOF( QUERY( fcr <* SELF\composite_shape_aspect.component_relationships |
          ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
```

```
            TYPEOF(fcr.related_shape_aspect)) )))
          )) = 1 )
        )) = 1 ));
END_ENTITY;
(*
```

Formal propositions:

WR1: Each instance of **compound_feature_in_solid** shall either be of type **instanced_feature** and **composite_shape_aspect**, or shall have a shape that has exactly one **shape_aspect** identified on it that is of type **composite_shape_aspect.**

WR2: The **shape_aspect_relationships** that are used to relate the single features with the **composite_shape_aspect** shall be **feature_component_relationships.**

WR3: If the **compound_feature_in_solid** is of type **instanced_feature** and **composite_shape_aspect** the **related_shape_aspects** defined by its **component_relationships** shall be of type **instanced_feature**, **transition_feature**, or **thread_occurrence.**

WR4: If the **compound_feature_in_solid** is not of type **instanced_feature**, the **composite_shape_aspect** identified on its shape shall only have **related_shape_aspects** defined by its **component_relationships** that are of type **placed_feature**, **transition_feature**, or **thread_occurrence.**

WR5: Each **placed_feature** that is specified by the **component_relationships** of a **composite_shape_aspect** of a **compound_feature_in_solid** shall have a **representation_item** of type **placement** with a **name** of 'element placement' in its **shape_representation_with_parameters.**

5.2.3.1.40 compound_shape_representation

A **compound_shape_representation** is a type of **shape_representation** that represents the shape or a portion of the shape of a product using elements of different dimensionality such as wireframes or surface models that are topologically connected. The topology of a **compound_shape_representation** may be non-manifold. See ARM definition for Compound_model in paragraph 4.2.87 for more information.

EXPRESS specification:

```
*)
ENTITY compound_shape_representation
  SUBTYPE OF (shape_representation);
WHERE
  WR1:  ('AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_CONTEXT' IN
         TYPEOF(SELF.context_of_items)) AND
        (SELF.context_of_items\geometric_representation_context.
              coordinate_space_dimension = 3);
  WR2:  SIZEOF( QUERY( cbsr_i <* SELF.items |
                SIZEOF( ['AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_MODEL',
```

```
                        'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL',
                        'AUTOMOTIVE_DESIGN.MAPPED_ITEM',
                        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D'] *
                       TYPEOF(cbsr_i)
               ) <> 1
        )) = 0;
  WR3:  SIZEOF( QUERY( cbsr_i <* SELF.items |
               SIZEOF( ['AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_MODEL',
                        'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL',
                        'AUTOMOTIVE_DESIGN.MAPPED_ITEM'] *
                       TYPEOF(cbsr_i)
               ) = 1
        )) > 0;
  WR4:  SIZEOF( QUERY( cbsr_i <* SELF.items |
             ('AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(cbsr_i)) AND
             (SIZEOF(['AUTOMOTIVE_DESIGN.COMPOUND_SHAPE_REPRESENTATION',
                      'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION',
                      'AUTOMOTIVE_DESIGN.NON_MANIFOLD_SURFACE_SHAPE_REPRESENTATION'] *
                     TYPEOF(cbsr_i\mapped_item.mapping_source)
              ) <> 1 )
        )) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **compound_shape_representation** shall have a **coordinate_space_dimension** equal to three.

WR2: The **items** in the **compound_shape_representation** shall be of type **edge_based_wireframe_model, face_based_surface_model, mapped_item** or **axis2_placement_3d.**

WR3: The **items** of a **compound_shape_representation** shall include at least one **mapped_item** or one **edge_based_wireframe_model** or one **face_based_surface_model.**

WR4: If there is a **mapped_item** in a **compound_shape_representation,** the source of the **mapped_item** shall be a **compound_shape_representation,** a **edge_based_wireframe_shape_representation,** or a **non_manifold_surface_shape_representation.**

Informal propositions:

IP1: All elements in a **compound_shape_representation** shall be topologically connected.

5.2.3.1.41 configured_effective_context_assignment

A **configured_effective_context_assignment** is a type of **effective_context_assignment** that assigns an effectivity to a **configured_effective_context_item** that specifies the context for which the usage of a configuration controlled item is valid. See ARM definition for Configuration in paragraph 4.2.89, for Dated_configuration in paragraph 4.2.115, for Lot_configuration in paragraph 4.2.272, and for Serial_configuration in paragraph 4.2.393 for more information.

EXPRESS specification:

```
*)
ENTITY configured_effective_context_assignment
  SUBTYPE OF (effective_context_assignment);
  items : SET [1:?] OF configured_effective_context_item;
WHERE
  WR1:  SIZEOF(SELF.items) = 1;
  WR2:  SIZEOF( QUERY( ea <* USEDIN(SELF.assigned_effectivity,
              'AUTOMOTIVE_DESIGN.EFFECTIVITY_ASSIGNMENT.' +
              'ASSIGNED_EFFECTIVITY') |
         'AUTOMOTIVE_DESIGN.CONFIGURED_EFFECTIVE_ITEM_ASSIGNMENT' IN
          TYPEOF(ea)
               )) = 1;
  WR3:  SELF.role.name = 'configuration';
END_ENTITY;
(*
```

Attribute definitions:

items: The set of items that specify the valid context for the **assigned_effectivity.**

Formal propositions:

WR1: The **configured_effective_context_assignment** shall contain exactly one item in its set of **items**.

WR2: The **configured_effective_context_assignment** shall reference as its **assigned_effectivity** an **effectivity** that is referenced as the **assigned_effectivity** by exactly one **configured_effective_item_assignment.**

WR3: The **configured_effective_context_assignment** shall reference as its **role** an **effectivity_role** that has a **name** of 'configuration'.

5.2.3.1.42 configured_effective_item_assignment

A **configured_effective_item_assignment** is a type of **effective_item_assignment** that assigns an effectivity to a **configured_effective_item** with the purpose of applying configuration control mechanisms to it. See ARM definition for Configuration in paragraph 4.2.89, for Dated_configuration in paragraph 4.2.115, for Lot_configuration in paragraph 4.2.272, and for Serial_configuration in paragraph 4.2.393 for more information.

EXPRESS specification:

```
*)
ENTITY configured_effective_item_assignment
  SUBTYPE OF (effective_item_assignment);
  items : SET [1:?] OF configured_effective_item;
WHERE
```

```
  WR1:  SIZEOF(SELF.items) = 1;
  WR2:  SIZEOF( QUERY( ae <* USEDIN(SELF.assigned_effectivity,
              'AUTOMOTIVE_DESIGN.EFFECTIVITY_ASSIGNMENT.' +
              'ASSIGNED_EFFECTIVITY') |
         'AUTOMOTIVE_DESIGN.CONFIGURED_EFFECTIVE_CONTEXT_ASSIGNMENT' IN
          TYPEOF(ae)
              )) = 1;
  WR3:  SELF.role.name = 'configuration';
END_ENTITY;
(*
```

Attribute definitions:

items: The set of items to which the effectivity is assigned.

Formal propositions:

WR1: The **configured_effective_item_assignment** shall contain exactly one item in its set of **items**.

WR2: The **configured_effective_item_assignment** shall reference as its **assigned_effectivity** an **effectivity** that is referenced as the **assigned_effectivity** by exactly one **configured_effective_context_assignment.**

WR3: The **configured_effective_item_assignment** shall reference as its **role** an **effectivity_role** that has a **name** of 'configuration'.

5.2.3.1.43 connected_face_shape_representation

A **connected_face_shape_representation** is a type of **representation** that represents a portion of a part's boundary by a connected set of faces.

EXPRESS specification:

```
*)
ENTITY connected_face_shape_representation
  SUBTYPE OF (representation);
WHERE
  WR1:  SIZEOF(SELF.items) = 1;
  WR2:  SIZEOF( QUERY( i <* SELF.items |
                ('AUTOMOTIVE_DESIGN.CLOSED_SHELL' IN TYPEOF(i)) OR
                ('AUTOMOTIVE_DESIGN.OPEN_SHELL' IN TYPEOF(i))
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **connected_face_shape_representation** shall have exactly one **representation_item** in its set of **items**.

WR2: The geometric element that is used to represent the **connected_face_shape_representation** shall be a **closed_shell** or an **open_shell.**

5.2.3.1.44 constructive_geometry_representation

A **constructive_geometry_representation** is a type of **representation** that is a collection of non-shape geometry that is used to support the definition of the geometry that is representing the shape of the part. See ARM definition for Constructive_geometry in paragraph 4.2.95 for more information.

EXPRESS specification:

```
*)
ENTITY constructive_geometry_representation
  SUBTYPE OF (representation);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_CONTEXT' IN
         TYPEOF(SELF.context_of_items);
  WR2:  SIZEOF( QUERY( cgr_i <* SELF.items |
                SIZEOF(['AUTOMOTIVE_DESIGN.PLACEMENT',
                        'AUTOMOTIVE_DESIGN.CURVE',
                        'AUTOMOTIVE_DESIGN.EDGE',
                        'AUTOMOTIVE_DESIGN.FACE',
                        'AUTOMOTIVE_DESIGN.POINT',
                        'AUTOMOTIVE_DESIGN.SURFACE',
                        'AUTOMOTIVE_DESIGN.VERTEX'] * TYPEOF(cgr_i)) <> 1
        )) = 0;
  WR3:  SIZEOF( QUERY( rr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'REPRESENTATION_RELATIONSHIP.REP_2') |
                'AUTOMTOIVE_DESIGN.CONSTRUCTIVE_GEOMETRY_REPRESENTATION_RELATIONSHIP' IN
                 TYPEOF(rr)
              )) < 1;
  WR4:  SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'REPRESENTATION_MAP.MAPPED_REPRESENTATION')
        ) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **constructive_geometry_representation** shall have a **geometric_representation_context** as **context_of_items.**

WR2: The **items** in the **constructive_geometry_representation** shall be of type **placement**, **curve**, **edge**, **face**, **point**, **surface**, or **vertex**.

WR3: The **constructive_geometry_representation** shall play the role of **rep_2** for at least one **constructive_geometry_representation_relationship.**

WR4: The **constructive_geometry_representation** shall not play the role of **mapped_representation** for a **representation_map.**

5.2.3.1.45 constructive_geometry_representation_relationship

A **constructive_geometry_representation_relationship** is a type of **representation_relationship** that associates the geometry representing the part's shape with the constructive geometry which was used for its definition or a constructive geometry with another constructive geometry which was used for its definition.

EXPRESS specification:

```
*)
ENTITY constructive_geometry_representation_relationship
  SUBTYPE OF (representation_relationship);
WHERE
  WR1:  SELF.rep_1.context_of_items :=: SELF.rep_2.context_of_items;
  WR2:  'AUTOMOTIVE_DESIGN.CONSTRUCTIVE_GEOMETRY_REPRESENTATION' IN
         TYPEOF(SELF.rep_2);
  WR3:  SIZEOF(['AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION',
                 'AUTOMOTIVE_DESIGN.CONSTRUCTIVE_GEOMETRY_REPRESENTATION'] *
                TYPEOF(SELF.rep_1)
        ) = 1;
  WR4:  NOT('AUTOMOTIVE_DESIGN.REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION' IN
            TYPEOF(SELF));
END_ENTITY;
(*
```

Formal propositions:

WR1: Both **representation**s playing the roles of **rep_1** and **rep_2** of the **constructive_geometry_representation_relationship** shall share the same **geometric_representation_context.**

WR2: The **representation** playing the role of **rep_2** of the **constructive_geometry_representation_relationship** shall be of type **constructive_geometry_representation.**

WR3: The **representation** playing the role of **rep_1** of the **constructive_geometry_representation_relationship** shall be of type **shape_representation** or **constructive_geometry_representation.**

WR4: The **constructive_geometry_representation_relationship** shall not be of type **representation_relationship_with_transformation.**

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **constructive_geometry_representation_relationship** entity:

– restrict_multi_language_for_constructive_geometry_representation_relationship (see 5.2.4.98).

5.2.3.1.46 default_tolerance_table

A **default_tolerance_table** is a type of **representation** that specifies information about general tolerances. See ARM definition for General_tolerance_table in paragraph 4.2.207 for more information.

EXPRESS specification:

```
*)
ENTITY default_tolerance_table
  SUBTYPE OF (representation);
WHERE
  WR1:  SIZEOF( QUERY( rep_rel <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
         'REPRESENTATION_RELATIONSHIP.REP_1') |
         'AUTOMOTIVE_DESIGN.DEFAULT_TOLERANCE_TABLE_CELL'
         IN TYPEOF(rep_rel.rep_2)
        )) > 0;
  WR2:  SELF\representation.context_of_items.context_type = 'default setting';
END_ENTITY;
(*
```

Formal propositions:

WR1: Each **default_tolerance_table** shall participate in at least one **representation_relationship** in the role of **rep_1** where the **representation** that plays the role of **rep_2** is of type **default_tolerance_table_cell**.

WR2: The **context_of_items** of the **default_tolerance_table** shall reference a **representation_context** with a **context_type** of 'default setting'.

5.2.3.1.47 default_tolerance_table_cell

A **default_tolerance_table_cell** is a type of **representation** that specifies one cell entry in a **default_tolerance_table.** See ARM definition for Tolerance_table_cell in paragraph 4.2.462 for more information.

EXPRESS specification:

```
*)
ENTITY default_tolerance_table_cell
  SUBTYPE OF (representation);
WHERE
```

```
WR1:  SIZEOF( QUERY( rep_rel <* USEDIN( SELF,
             'AUTOMOTIVE_DESIGN.REPRESENTATION_' +
             'RELATIONSHIP.REP_2') |
       'AUTOMOTIVE_DESIGN.DEFAULT_TOLERANCE_TABLE'
       IN TYPEOF(rep_rel.rep_1)
          )) = 1;
WR2:  SELF\representation.context_of_items.context_type = 'default setting';
WR3:  ((SIZEOF( QUERY( item <* SELF\representation.items |
        ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
                'ITEM' IN TYPEOF(item))
       AND
          (item\representation_item.name = 'significant number of digits')
          )) = 1)
       AND
        (SIZEOF( QUERY( item <* SELF\representation.items |
          ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
          'ITEM' IN TYPEOF(item))
       AND
          (SIZEOF( [item\representation_item.name] * ['lower limit',
        'upper limit']) > 0)
          )) = 0)
           )

       OR
           (
            (SIZEOF( QUERY( item <* SELF\representation.items |
          ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
          'ITEM' IN TYPEOF(item))
       AND
          (item\representation_item.name = 'significant number of digits')
          )) = 0)
       AND
            (SIZEOF( QUERY( item <* SELF\representation.items |
          ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
       'ITEM' IN TYPEOF(item))
       AND
          (item\representation_item.name = 'lower limit')
          )) = 1)
       AND
        (SIZEOF( QUERY( item <* SELF\representation.items |
          ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
          'ITEM' IN TYPEOF(item))
       AND
          (item\representation_item.name = 'upper limit')
          )) = 1)
           );
WR4:  ((SIZEOF( QUERY( item <* SELF\representation.items |
          ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
          'ITEM' IN TYPEOF(item))
       AND
          (item\representation_item.name = 'plus minus tolerance value')
          )) = 1)
```

```
         AND
          (SIZEOF( QUERY( item <* SELF\representation.items |
            ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
            'ITEM' IN TYPEOF(item))
         AND
            (SIZEOF( [item\representation_item.name] * ['lower tolerance value',
          'upper tolerance value']) > 0)
            )) = 0)
             )

         OR
             (
              (SIZEOF( QUERY( item <* SELF\representation.items |
            ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
            'ITEM' IN TYPEOF(item))
         AND
            (item\representation_item.name = 'plus minus tolerance value')
             )) = 0)
         AND
              (SIZEOF( QUERY( item <* SELF\representation.items |
            ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
            'ITEM' IN TYPEOF(item))
         AND
            (item\representation_item.name = 'lower tolerance value')
            )) = 1)
         AND
          (SIZEOF( QUERY( item <* SELF\representation.items |
            ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
            'ITEM' IN TYPEOF(item))
         AND
            (item\representation_item.name = 'upper tolerance value')
            )) = 1)
             );
  WR5:  SIZEOF(SELF.items) <= 4;
END_ENTITY;
(*
```

Formal propositions:

WR1: Each **default_tolerance_table_cell** shall participate in exactly one **representation_relationship** in the role of the **rep_2** where a **default_tolerance_table** participates in the role of **rep_1**.

WR2: The **context_of_items** of the **default_tolerance_table** shall reference a **representation_context** with a **context_type** of 'default setting'.

WR3: Either exactly one **representation_item** in the set of **items** shall be a **measure_representation_item** with the **name** of 'significant number of digits' or there shall be two **representation_items** of type **measure_representation_item** where one of them has the **name** of 'lower limit' and the other one has the **name** of 'upper limit'.

WR4: Either exactly one **representation_item** in the set of **items** shall be a **measure_representation_item** with the **name** of 'plus minus tolerance value' or there shall be two **representation_items** of type **measure_representation_item** where one of them has the **name** of 'lower tolerance value' and the other one has the **name** of 'upper tolerance value'

WR5: The **default_tolerance_table_cell** shall contain at most four **representation_items** in its set of **items**.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **default_tolerance_table_cell** entity:

– restrict_multi_language_for_default_tolerance_table_cell (see 5.2.4.100).

5.2.3.1.48 digital_document_reference

A **digital_document_reference** is a type of **document_reference** that assigns a document that contains the externally defined geometry information to an **externally_defined_representation.** See ARM definition for External_model in paragraph 4.2.166 for more information.

EXPRESS specification:

```
*)
ENTITY digital_document_reference
  SUBTYPE OF (document_reference);
  items : SET [1:?] OF digital_document_reference_item;
WHERE
  WR1:  SELF.assigned_document.kind.product_data_type = 'digital';
  WR2:  SELF.role.name = 'external definition';
  WR3:  SIZEOF(SELF.items) = 1;
END_ENTITY;
(*
```

Attribute definitions:

items: the set of **items** to which the digital document is assigned.

Formal propositions:

WR1: The **digital_document_reference** shall reference as its **assigned_document** a **document** that references as its **kind** a **document_type** with a **product_data_type** of 'digital'.

WR2: The **digital_document_reference** shall reference as its **role** a **document_role** that has a **name** of 'external definition'.

WR3: The **digital_document_reference** shall contain exactly one item in its set of **items**.

5.2.3.1.49 directed_angle

A **directed_angle** is a type of **shape_aspect** that specifies an angle value and the element the angle is measured from. See ARM definition for Angle_specification in paragraph 4.2.9 for more information.

EXPRESS specification:

```
*)
ENTITY directed_angle
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
        TYPEOF(SELF.of_shape.definition);
  WR2:  SIZEOF( QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF (pdr.used_representation)
              )) = 1;
  WR3:  SIZEOF( QUERY (pdr <* USEDIN (SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF (pdr.used_representation)) AND
         (SIZEOF(pdr.used_representation.items) = 2)
              )) = 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
         (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
          ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
            TYPEOF(srwp_i)) AND
          (srwp_i.name = 'angle reference') AND
          (srwp_i.description IN ['punch direction',
                                                 'profile normal','surface normal'])
                 )) = 1 )
              )) = 1 ;
  WR5:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
         (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
          (srwp_i.name = 'angle') AND
          (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
           'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
          * TYPEOF(srwp_i)) = 2 )
                 )) = 1 )
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **directed_angle** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **directed_angle** shall have exactly one implicit representation defined by a **shape_representation_with_parameters.**

WR3: The **shape_representation_with_parameters** that represents the shape of the **directed_angle** shall contain exactly two **representation_items** in its set of **items**.

WR4: The implicit representation of the **directed_angle** shall have exactly one **representation_item** in its set of **items** which is a **descriptive_representation_item** with a **name** of 'angle reference' and a **description** of either 'punch direction', 'profile normal', or 'surface normal'.

WR5: The implicit representation of the **directed_angle** shall have exactly one **representation_item** in its set of **items** which is of type **measure_representation_item** and **plane_angle_measure_with_unit** and has a **name** of 'angle'.

5.2.3.1.50 directed_dimensional_location

A **directed_dimensional_location** is a type of **dimension_location** that identifies the direction to measure the location dimension.

NOTE – The definition of this entity is identical to the definition in ISO 10303–224.

EXPRESS specification:

```
*)
ENTITY directed_dimensional_location
  SUBTYPE OF (dimensional_location);
END_ENTITY;
(*
```

Attribute definitions:

SELF\shape_aspect_relationship.related_shape_aspect: the target of the directed dimension.

SELF\shape_aspect_relationship.relating_shape_aspect: the origin of the directed dimension.

5.2.3.1.51 direction_shape_representation

A **direction_shape_representation** is a type of **representation** that represents a direction vector in two or three dimensional space.

EXPRESS specification:

```
*)
ENTITY direction_shape_representation
  SUBTYPE OF (representation);
WHERE
  WR1:  SIZEOF(SELF.items) = 1;
  WR2:  SIZEOF( QUERY( i <* SELF.items |
         'AUTOMOTIVE_DESIGN.DIRECTION' IN TYPEOF(i)
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **direction_shape_representation** shall have exactly one **representation_item** in its set of items.

WR2: The geometric element that is used to represent the **direction_shape_representation** shall be a **direction**.

5.2.3.1.52 draughting_model

A **draughting_model** is a type of **representation** that represents of the shape of a product for the purpose of draughting. A **draughting_model** may contain **mapped_item**s that are the mappings of **shape_representation**s or of other **draughting_model**s. A **mapped_item** that is the mapping of a **shape_representation** specifies an explicit representation of the shape of a product that is to be presented in the drawing. A **mapped_item** that is the mapping of another **draughting_model** specifies the use of a sub-model.
The **draughting_model** may contain **styled_item**s. These **styled_item**s specify the style for a **mapped_item** that is the mapping of a **shape_representation,** and provide the **curve_style** to be used in presenting the **shape_representation** in the drawing.The **draughting_model** may contain **annotation_occurrence**s. These **annotation_occurrence**s specify annotation that is placed in the same coordinate system as the **shape_representation,** and is projected into a view when the **draughting_model** is presented in a drawing.

EXPRESS specification:

```
*)
ENTITY draughting_model
  SUBTYPE OF (representation);
UNIQUE
  UR1:  SELF\representation.name;
WHERE
  WR1:  SIZEOF (QUERY (it <* SELF.items |
          NOT (SIZEOF (['AUTOMOTIVE_DESIGN.MAPPED_ITEM',
                        'AUTOMOTIVE_DESIGN.STYLED_ITEM',
                        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT',
```

```
                        'AUTOMOTIVE_DESIGN.CAMERA_MODEL_D3'] *
              TYPEOF (it)) = 1
       ))) = 0;
  WR2:  SIZEOF (QUERY (mi <* QUERY (it <* SELF.items |
          ('AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(it))) |
          NOT (SIZEOF (['AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION',
                        'AUTOMOTIVE_DESIGN.DRAUGHTING_MODEL'] *
              TYPEOF (mi\mapped_item.mapping_source.
                         mapped_representation)) = 1
       ))) = 0;
  WR3:  SIZEOF (QUERY (smi <* QUERY (si <* QUERY (it <* SELF.items |
          ('AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it))) |
          ('AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN
              TYPEOF(si\styled_item.item))) |
          NOT (('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN
                   TYPEOF(smi\styled_item.item\mapped_item.
                         mapping_source.mapped_representation))
              AND
               (SIZEOF (QUERY (sty <* smi\styled_item.styles |
                 NOT (SIZEOF (QUERY (psa <* sty.styles |
                        NOT ('AUTOMOTIVE_DESIGN.CURVE_STYLE' IN
                                TYPEOF(psa)))) = 1
              ))) = 1))
       )) = 0;
END_ENTITY;
(*
```

Formal propositions:

UR1: The name of a **draughting_model** shall be unique.

WR1: Each item of a **draughting_model** shall be a **mapped_item**, **styled_item**, **axis2_placement**, or **camera_model_d3.**

WR2: If there is a **mapped_item** in the **draughting_model**, the source of the **mapped_item** shall be a **shape_representation** or another **draughting_model.**

WR3: If a **mapped_item** is styled, the source of the **mapped_item** shall be a **shape_representation** and the style applied to the **mapped_item** shall be a **curve_style.**

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **draughting_model** entity:

– draughting_model_annotation_layers (see 5.2.4.51);

– restrict_multi_language_for_draughting_model (see 5.2.4.106);

– restrict_version_assignment_for_draughting_model (see 5.2.4.171).

5.2.3.1.53 draughting_pre_defined_text_font

A **draughting_pre_defined_text_font** is a type of **pre_defined_text_font** that is identified by name. The definition of the appearance of each **draughting_pre_defined_text_font** is given in ISO 3098-1.

NOTE – The definition of this entity is identical to the definition in ISO 10303–202.

EXPRESS specification:

```
*)
ENTITY draughting_pre_defined_text_font
  SUBTYPE OF (pre_defined_text_font);
WHERE
  WR1:  SELF.name IN ['ISO 3098-1 font A', 'ISO 3098-1 font B'];
END_ENTITY;
(*
```

Formal propositions:

WR1: The name of the **draughting_pre_defined_text_font** shall be 'ISO 3098-1 font A' or 'ISO 3098-1 font B'.

Attribute value definitions:

The **draughting_pre_defined_text_font**s are:

ISO 3098-1 font A: the text font denoted as Lettering A in clause 3 of ISO 3098-1.

ISO 3098-1 font B: the text font denoted as Lettering B in clause 3 of ISO 3098-1.

5.2.3.1.54 draughting_specification_reference

A **draughting_specification_reference** is a type of **document_reference** that is the designation of an external draughting specification for a drawing revision.

NOTE – The definition of this entity is identical to the definition in ISO 10303–505.

EXPRESS specification:

```
*)
ENTITY draughting_specification_reference
  SUBTYPE OF (document_reference);
  specified_items : SET [1:?] OF specified_item;
WHERE
  WR1:  (SELF.assigned_document.kind.product_data_type =
                   'draughting specification');
END_ENTITY;
(*
```

Attribute definitions:

specified_items: The **drawing_revision**s to which a draughting specification is applied.

Formal propositions:

WR1: Each document assigned shall be a draughting specification.

5.2.3.1.55 edge_round

An **edge_round** is a type of **transition_feature** that is the representation of a circular convex transition between two faces. See ARM definition for Edge_round in paragraph 4.2.154 for more information.

EXPRESS specification:

```
*)
ENTITY edge_round
  SUBTYPE OF (transition_feature);
WHERE
  WR1:  (SELF.description <> 'constant radius') XOR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)
               )) > 0 );
  WR2:  (SELF.description <> 'constant radius') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                 ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          ({1 <= SIZEOF(pdr.used_representation.items) <= 3})
               )) = 1 );
  WR3:  (SELF.description <> 'constant radius') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
           srwp_i.name IN ['radius','first face offset',
                      'second face offset']
         )) = SIZEOF(pdr.used_representation.items))
               )) = 1 );
  WR4:  (SELF.description <> 'constant radius') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
```

```
                    (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
            'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 ) AND
           (srwp_i.name = 'radius')
         )) = 1 )
               )) = 1 );
  WR5:  (SELF.description <> 'constant radius') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
           (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
            'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 ) AND
           (srwp_i.name = 'first face offset')
         )) <= 1 )
               )) = 1 );
  WR6:  (SELF.description <> 'constant radius') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
             (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
            'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 ) AND
           (srwp_i.name = 'second face offset')
         )) <= 1 )
          )) = 1 );
END_ENTITY;
(*
```

Formal propositions:

WR1: An **edge_round** has an implicit representation if and only if it has a **description** of 'constant radius'.

WR2: If the **description** of the **edge_round** is 'constant radius', the implicit representation shall contain between one and three **representation_items** in its set of **items**.

WR3: If the **description** of the **edge_round** is 'constant radius', the implicit representation shall only contain **representation_items** in its set of **items** that have a **name** of either 'radius', 'first face offset', or 'second face offset'.

WR4: If the **description** of the **edge_round** is 'constant radius', exactly one **representation_item** used for the implicit representation of the **edge_round** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR5: If the **description** of the **edge_round** is 'constant radius', at most one **representation_item** used for the implicit representation of the **edge_round** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'first face offset'.

WR6: If the **description** of the **edge_round** is 'constant radius', at most one **representation_item** used for the implicit representation of the **edge_round** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'second face offset'.

5.2.3.1.56 element_delivery

An **element_delivery** is a type of **directed_action** that specifies the delivery of a physical object defined by either a **product_definition_formation,** by a **configurable_item** that identifies a physically realizable object by selecting a set of **product_concept_feature** instances for a product class, or by a **product_definition** that identifies a physically realized occurrence of a product. See ARM definition for Element_delivery in paragraph 4.2.158 for more information.

EXPRESS specification:

```
*)
ENTITY element_delivery
  SUBTYPE OF (directed_action);
WHERE
  WR1:  SIZEOF( QUERY( aa <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
         'AUTOMOTIVE_DESIGN.ELEMENT_DELIVERY_ASSIGNMENT' IN
          TYPEOF(aa)
              )) = 1;
  WR2:  (SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')) = 1) AND
        (SIZEOF( QUERY( aoa <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
          aoa.role.name <> 'destination'
         )) = 0 );
  WR3:  (SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'ACTION_PROPERTY.DEFINITION')) <= 1 ) AND
        ((SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
         'ACTION_PROPERTY.DEFINITION')) = 0) OR
         (SIZEOF( QUERY( ap <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'ACTION_PROPERTY.DEFINITION') |
         SIZEOF( USEDIN( ap, 'AUTOMOTIVE_DESIGN.' +
             'ACTION_PROPERTY_REPRESENTATION.PROPERTY')) <> 1
          )) = 0 ) AND
         (SIZEOF( QUERY( ap <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'ACTION_PROPERTY.DEFINITION') |
         SIZEOF( QUERY( apr <* USEDIN( ap, 'AUTOMOTIVE_DESIGN.' +
               'ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
         (apr.representation.description <> 'delivery quantity') OR
         (SIZEOF(apr.items) <> 1) OR
         (SIZEOF( QUERY( apr_i <* apr.items |
          NOT ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN
               TYPEOF(apr_i)) OR
          (apr_i\measure_representation_item.name <> 'quantity')
               )) > 0 )
              )) > 0
```

```
        )) = 0 ));
END_ENTITY;
(*
```

Formal propositions:

WR1: The **element_delivery** shall be referenced as the **assigned_action** by exactly one instance of **element_delivery_assignment.**

WR2: There shall be exactly one instance of **applied_organization_assignment** that contains the **element_delivery** in its set of **items**. This **applied_organization_assignment** shall reference as its **role** an **organization_role** that has a **name** of 'destination'.

WR3: There shall be at most one instance of **action_property** that references the **element_delivery** as its **definition**. If present, this **action_property** shall be referenced as the **property** by exactly one instance of **action_property_representation.** This **action_property_representation** shall reference as its **representation** a **representation** that has a **description** of 'delivery quantity' and that contains exactly one **representation_item** in its set of **items**. This **representation_item** shall be a **measure_representation_item** and shall have a **name** of 'quantity'.

Informal propositions:

IP1: In case the **element_delivery** is not referenced by an **action_property** as the **definition**, the quantity of the delivery shall be one.

5.2.3.1.57 element_delivery_assignment

An **element_delivery_assignment** is a type of **action_assignment** that assigns an **element_delivery** to an item to be delivered. See ARM definition for Element_delivery in paragraph 4.2.158 for more information.

EXPRESS specification:

```
*)
ENTITY element_delivery_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF element_delivery_item;
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.ELEMENT_DELIVERY' IN TYPEOF(SELF.assigned_action);
  WR2:  SELF.role.description = 'deliverable element';
  WR3:  (SIZEOF(SELF.items) = 1 ) AND
        (SIZEOF( QUERY( i <* SELF.items |
          NOT ( 'AUTOMOTIVE_DESIGN.APPLIED_ACTION_ASSIGNMENT' IN
                TYPEOF(i)) OR
          (i\applied_action_assignment.role.description <> 'activity element') OR
          (SIZEOF(i\applied_action_assignment.items) <> 1) OR
          (SIZEOF( QUERY( i_i <* i\applied_action_assignment.items |
```

```
          'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_FORMATION' IN
           TYPEOF(i_i)
          )) +
          SIZEOF( QUERY( i_i <* i\applied_action_assignment.items |
          ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i_i)) AND
          (i_i.name = 'physical instance') AND
          (i_i.frame_of_reference.name = 'physical occurrence')
          )) +
          SIZEOF( QUERY( i_i <* i\applied_action_assignment.items |
          ('AUTOMOTIVE_DESIGN.CONFIGURABLE_ITEM' IN TYPEOF(i_i)) AND
          (i_i.description = 'product specification')
          )) <> 1 )
        )) = 0 );
END_ENTITY;
(*
```

Attribute definitions:

items: The set of items to which the **element_delivery** is assigned.

Formal propositions:

WR1: The **element_delivery_assignment** shall reference as its **assigned_action** an action that is of type **element_delivery.**

WR2: The **element_delivery_assignment** shall reference as its **role** an **action_role** that has a **description** of 'deliverable element'.

WR3: The **element_delivery_assignment** shall contain exactly one item in its set of **items**. This item shall be an **applied_action_assignment** that references as its **role** an **action_role** that has a **description** of 'activity element' and that contains exactly one item in its set of **items**. This item shall either be a **product_definition_formation,** a **product_definition** that has a **name** of 'physical instance' and that references as its **frame_of_reference** a **product_definition_context** that has a **name** of 'physical occurrence', or a **configurable_item** that has a **description** of 'product specification'.

5.2.3.1.58 event_assignment

An **event_assignment** is a type of **action_assignment** that assigns an event to an **event_item.** See ARM definition for Event_reference in paragraph 4.2.160 for more information.

EXPRESS specification:

```
*)
ENTITY event_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF event_item;
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.EVENT_REFERENCE' IN TYPEOF(SELF.assigned_action);
```

```
  WR2:  (SIZEOF( QUERY( i <* SELF.items |
          'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY' IN TYPEOF(i)
         )) = 0 ) OR
        ((SIZEOF( QUERY( i <* SELF.items |
           'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY' IN TYPEOF(i)
          )) > 0 ) AND
         (SELF.role.name IN ['effectivity start',
              'effectivity end']));
  WR3:  (SIZEOF( QUERY( i <* SELF.items |
          'AUTOMOTIVE_DESIGN.PROJECT' IN TYPEOF(i)
         )) = 0 ) OR
        ((SIZEOF( QUERY( i <* SELF.items |
           'AUTOMOTIVE_DESIGN.PROJECT' IN TYPEOF(i)
          )) > 0 ) AND
         (SELF.role.name IN ['actual start',
              'planned start',
              'planned end']));
END_ENTITY;
(*
```

Attribute definitions:

items: The set of items to which the **event_reference** is assigned.

Formal propositions:

WR1: The **event_assignment** shall reference as its **assigned_action** an action that is of type **event_reference**.

WR2: If the **event_assignment** contains an **event_based_effectivity** in its set of **items**, it shall reference as its **role** an **action_role** that has a **name** of either 'effectivity start' or 'effectivity end'.

WR3: If the **event_assignment** contains a **project** in its set of **items**, it shall reference as its **role** an **action_role** that has a **name** of either 'actual start', 'planned start', or 'planned end'.

5.2.3.1.59 event_based_effectivity

An **event_based_effectivity** is a type of effectivity that specifies the start or end of the valid use of an aspect of product data by referring to an event. See ARM definition for Effectivity in paragraph 4.2.156 for more information.

EXPRESS specification:

```
*)
ENTITY event_based_effectivity
  SUBTYPE OF (effectivity);
WHERE
  WR1:  SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT.ITEMS')) > 0;
```

```
  WR2:  SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT.ITEMS')) <= 2;
  WR3:  SIZEOF( QUERY( ea <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT.ITEMS') |
         ea.role.name = 'effectivity start' )) <= 1;
  WR4:  SIZEOF( QUERY( ea <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT.ITEMS') |
         ea.role.name = 'effectivity end' )) <= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **event_based_effectivity** shall be contained in the set of **items** of at least one **event_assignment.**

WR2: The **event_based_effectivity** shall be contained in the set of **items** of at most two **event_assignmen**s.

WR3: The **event_based_effectivity** shall be contained in the set of **items** of at most one **event_assignment** that references as its **role** an **action_role** with a **name** of 'effectivity start'.

WR4: The **event_based_effectivity** shall be contained in the set of **items** of at most one **event_assignment** that references as its **role** an **action_role** with a **name** of 'effectivity end'.

5.2.3.1.60 event_context_assignment

An **event_context_assignment** is a type of **action_assignment** that assigns an **event_reference** to an **event_context_item** that specifies the context for the event. See ARM definition for Event_reference in paragraph 4.2.160 for more information.

EXPRESS specification:

```
*)
ENTITY event_context_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF event_context_item;
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.EVENT_REFERENCE' IN TYPEOF(SELF.assigned_action);
  WR2:  SELF.role.name = 'event context';
  WR3:  SIZEOF(SELF.items) = 1;
END_ENTITY;
(*
```

Attribute definitions:

items: The set of items that specify the context for the **event_reference.**

Formal propositions:

WR1: The **event_context_assignment** shall reference as its **assigned_action** an **action** that is of type **event_reference**.

WR2: The **event_context_assignment** shall reference as its **role** an **action_role** that has a **name** of 'event context'.

WR3: The **event_context_assignment** shall contain exactly one item in its set of **items**.

5.2.3.1.61 event_reference

An **event_reference** is a type of action that defines a point in time established with respect to an action. See ARM definition for Event_reference in paragraph 4.2.160 for more information.

EXPRESS specification:

```
*)
ENTITY event_reference
  SUBTYPE OF (action);
WHERE
  WR1:  SIZEOF( QUERY( aa <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
               'AUTOMOTIVE_DESIGN.EVENT_CONTEXT_ASSIGNMENT' IN
                TYPEOF(aa)
              )) <= 1;
  WR2:  (SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                 'ACTION_PROPERTY.DEFINITION')) <= 1 ) AND
        ((SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
         'ACTION_PROPERTY.DEFINITION')) = 0 ) OR
         (SIZEOF( QUERY( ap <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'ACTION_PROPERTY.DEFINITION') |
         (SIZEOF( USEDIN( ap, 'AUTOMOTIVE_DESIGN.' +
          'ACTION_PROPERTY_REPRESENTATION.PROPERTY')) = 1 ) AND
         (SIZEOF( QUERY( apr <* USEDIN( ap, 'AUTOMOTIVE_DESIGN.' +
               'ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
                (apr.representation.name <> 'offset definition') OR
                (SIZEOF( apr.representation.items) <> 1) OR
                (SIZEOF( QUERY( apr_r_i <* apr.representation.items |
               (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                'AUTOMOTIVE_DESIGN.TIME_MEASURE_WITH_UNIT']
               * TYPEOF(apr_r_i)) <> 2 ) OR
               (apr_r_i.name <> 'offset value')
               )) > 0 )
         )) = 0 )
          )) = 1 ));
END_ENTITY;
(*
```

Formal propositions:

WR1: The **event_reference** shall be referenced as the **assigned_action** by at most one instance of **event_context_assignment.**

WR2: There shall be at most one instance of **action_property** that references the **event_reference** as its **definition**. If present, this **action_property** shall be referenced as the **property** by exactly one instance of **action_property_representation.** This **action_property_representation** shall reference as its **representation** a **representation** that has a **description** of 'offset definition' and that contains exactly one **representation_item** in its set of **items**. This **representation_item** shall be of type **measure_representation_item** and **time_measure_with_unit** and shall have a **name** of 'offset value'.

5.2.3.1.62 exclusive_product_concept_feature_category

An **exclusive_product_concept_feature_category** is a type of **product_concept_feature_category** in which the contained **product_concept_feature** instances describe mutually exclusive characteristics of a product. See ARM definition for Specification_category in paragraph 4.2.416 for more information.

EXPRESS specification:

```
*)
ENTITY exclusive_product_concept_feature_category
  SUBTYPE OF (product_concept_feature_category);
END_ENTITY;
(*
```

5.2.3.1.63 externally_defined_classification_attribute

An **externally_defined_classification_attribute** is a type of representation and of **externally_defined_item** that represents a characteristic used to classify a product. The classification characteristic is specified in an external source. See ARM definition for Classification_attribute in paragraph 4.2.69 for more information.

EXPRESS specification:

```
*)
ENTITY externally_defined_classification_attribute
  SUBTYPE OF (externally_defined_item, representation);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.KNOWN_SOURCE' IN TYPEOF(SELF.source);
  WR2:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
                 (pdr.definition.description <> 'category attribute') OR
                 NOT ('AUTOMOTIVE_DESIGN.CHARACTERIZED_PRODUCT_CATEGORY' IN
                      TYPEOF(pdr.definition\property_definition.definition))
               )) = 0;
```

```
  WR3:  SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')) >= 1;
  WR4:  SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'REPRESENTATION_RELATIONSHIP.REP_1')) >= 1;
  WR5:  SIZEOF( QUERY( rr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'REPRESENTATION_RELATIONSHIP.REP_1') |
                (rr.name <> 'classification attribute value') OR
                NOT ('AUTOMOTIVE_DESIGN.PROPERTY_VALUE_REPRESENTATION' IN
                     TYPEOF(rr.rep_2))
              )) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **externally_defined_classification_attribute** shall reference as its **source** an **external_source** that is of type **known_source.**

WR2: The **externally_defined_classification_attribute** shall be referenced as the **used_representation** only by instances of **property_definition_representation** that reference as their **definition** a **property_definition** that has a **description** of 'category attribute' and that references as its **definition** a **characterized_object** of type **characterized_product_category.**

WR3: The **externally_defined_classification_attribute** shall be referenced as the **used_representation** by at least one instance of **property_definition_representation.**

WR4: The **externally_defined_classification_attribute** shall be referenced as the **rep_1** by at least one instance of **representation_relationship.**

WR5: The **externally_defined_classification_attribute** shall be referenced as the **rep_1** only by instances of **representation_relationship** that have a **name** of 'classification attribute value' and that reference as the **rep_2** a **property_value_representation.**

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **externally_defined_classification_attribute** entity:

– restrict_multi_language_for_externally_defined_classification_attribute (see 5.2.4.111).

5.2.3.1.64 externally_defined_classification_system

An **externally_defined_classification_system** is a type of **group** and of **externally_defined_item** that describes a method for classifying objects where the complete set of possible values is described in an external source. See ARM definition for Classification_system in paragraph 4.2.70 for more information.

EXPRESS specification:

```
*)
```

```
ENTITY externally_defined_classification_system
  SUBTYPE OF (externally_defined_item, group);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.KNOWN_SOURCE' IN TYPEOF(SELF.source);
END_ENTITY;
(*
```

Formal propositions:

WR1: The **externally_defined_classification_system** shall reference as its **source** an **external_source** that is of type **known_source.**

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **externally_defined_classification_system** entity:

– restrict_multi_language_for_externally_defined_classification_system (see 5.2.4.112).

5.2.3.1.65 externally_defined_product_category

An **externally_defined_product_category** is a type of **characterized_product_category** and of **externally_defined_item** that characterizes products of the same kind by referring to an entry in an external library due to ISO 13584. See ARM definition for General_classification in paragraph 4.2.192 for more information.

EXPRESS specification:

```
*)
ENTITY externally_defined_product_category
  SUBTYPE OF (characterized_product_category, externally_defined_item);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.KNOWN_SOURCE' IN TYPEOF(SELF.source);
END_ENTITY;
(*
```

Formal propositions:

WR1: The **externally_defined_product_category** shall reference as its **source** an **external_source** that is of type **known_source.**

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **externally_defined_product_category** entity:

– restrict_multi_language_for_externally_defined_product_category (see 5.2.4.113).

5.2.3.1.66 externally_defined_representation

An **externally_defined_representation** is a type of **representation** that is the identification of a geometric model that is described in a digital document. See ARM definition for External_model in paragraph 4.2.166 for more information.

EXPRESS specification:

```
*)
ENTITY externally_defined_representation
  SUBTYPE OF (representation);
WHERE
  WR1:  SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'DIGITAL_DOCUMENT_REFERENCE.ITEMS')) = 1;
  WR2:  SIZEOF (SELF.items) = 1;
  WR3:  SIZEOF (QUERY (it <* items |
               SIZEOF (['AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_2D',
                        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D']
                       * TYPEOF (it)) = 1)) =1;
END_ENTITY;
(*
```

Formal propositions:

WR1: There shall be exactly one instance of **digital_document_reference** that contains the **externally_defined_representation** in its set of **items**.

WR2: The **externally_defined_representation** shall contain exactly one **representation_item** in its set of items.

WR3: The **externally_defined_representation** shall contain exactly one **representation_item** that is either an **axis2_placement_2d** or an **axis2_placement_3d.**

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **externally_defined_representation** entity:

– restrict_multi_language_for_externally_defined_representation (see 5.2.4.114).

5.2.3.1.67 externally_defined_thread

An **externally_defined_thread** is a thread whose specification is given in a document. See ARM definition for Catalog_thread in paragraph 4.2.48 for more information.

EXPRESS specification:

```
*)
ENTITY externally_defined_thread
  SUBTYPE OF (externally_defined_item, feature_definition);
```

```
WHERE
  WR1:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                        (SIZEOF(pdr.used_representation.items) = 2)
               )) = 1
         )) = 1;
  WR2:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
                        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
          ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
            TYPEOF(srwp_i)) AND
          (srwp_i.name = 'qualifier'))) = 1)
               )) = 1
         )) = 1;
  WR3:  (SIZEOF( QUERY( adr <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_DOCUMENT_REFERENCE.ITEMS') |
                 adr.role.name = 'specification' )) > 0 ) OR
        ('AUTOMOTIVE_DESIGN.KNOWN_SOURCE' IN TYPEOF(SELF.source));
END_ENTITY;
(*
```

Formal propositions:

WR1: The **shape_representation_with_parameters** that represents the **externally_defined_thread** shall contain exactly two **representation_items** in its set of **items**.

WR2: Exactly one **representation_item** used for the implicit representation of the **externally_defined_thread** shall be of type **descriptive_representation_item** with a **name** of 'qualifier'.

WR3: An **externally_defined_thread** shall either be referenced by an **applied_document_reference** with a **document_role** that has a **name** of 'specification' or its attribute **source** shall reference a **known_source.**

5.2.3.1.68 face_shape_representation

A **face_shape_representation** is a type of **representation** that represents a portion of a part's boundary by a topologically bounded surface.

EXPRESS specification:

```
*)
```

```
ENTITY face_shape_representation
  SUBTYPE OF (representation);
WHERE
  WR1:  SIZEOF (SELF.items) = 1;
  WR2:  SIZEOF( QUERY (it <* SELF.items |
                 'AUTOMOTIVE_DESIGN.FACE_SURFACE' IN TYPEOF (it)
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **face_shape_representation** shall have exactly one **representation_item** in its set of **items**.

WR2: The geometric element that is used to represent the **face_shape_representation** shall be a **face_surface.**

5.2.3.1.69 feature_component_definition

A **feature_component_definition** is a type of **characterized_object** that is used as a constituent of a preconceived form pattern.

EXPRESS specification:

```
*)
ENTITY feature_component_definition
  SUBTYPE OF (characterized_object);
WHERE
  WR1:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT.OF_SHAPE') |
         NOT ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(sa))
                )) > 1
              )) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: A **feature_component_definition** shall be the basis shape for exactly one **shape_aspect.**

5.2.3.1.70 feature_component_relationship

A **feature_component_relationship** is a type of **shape_aspect_relationship** that is used to relate features with implicitly represented components.

EXAMPLE 238 – A Chamfer may have the implicit chamfer offset amount defined with a **feature_component_relationship.**

EXPRESS specification:

```
*)
ENTITY feature_component_relationship
  SUBTYPE OF (shape_aspect_relationship);
END_ENTITY;
(*
```

5.2.3.1.71 feature_definition

A **feature_definition** is a type of **characterized_object** that is a preconceived form pattern. See ARM definition for Feature_definition in paragraph 4.2.175 for more information.

EXPRESS specification:

```
*)
ENTITY feature_definition
  SUBTYPE OF (characterized_object);
WHERE
  WR1:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)
               )) = 1
              )) = 1;
  WR2:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
         (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
          ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i)) AND
          (srwp_i.name = 'orientation') )) = 1 )
               )) = 1
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **feature_definition** shall have exactly one implicit **representation**.

WR2: Exactly one **representation_item** in the implicit **representation** of the **feature_definition** shall be a **placement** with a **name** of 'orientation. This **placement** shall define a reference location and orientation for the origin of the **feature_definition.**

5.2.3.1.72 feature_in_panel

A **feature_in_panel** is a type of **feature_definition** applicable to panel parts. A **feature_in_panel** consists of a ruled surface between two closed profiles one of which lies on the panel. See ARM definition for Form_feature_in_panel in paragraph 4.2.188 for more information.

EXPRESS specification:

```
*)
ENTITY feature_in_panel
  SUBTYPE OF (feature_definition);
WHERE
  WR1:  SELF.description IN ['pocket','boss','stairstep'];
  WR2:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
         (SIZEOF(pdr.used_representation.items) = 3 )
               )) = 1
              )) = 1;
  WR3:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
                (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(srwp_i)
                ) = 2 ) AND
                (srwp_i.name = 'base fillet radius')
               )) = 1 )
               )) = 1
              )) = 1;
  WR4:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
```

```
        TYPEOF(pdr.used_representation)) AND
               (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
               (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(srwp_i)
               ) = 2 ) AND
               (srwp_i.name = 'target fillet radius')
              )) = 1 )
              )) = 1
             )) = 1;
WR5:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMTOIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
        (sa.name = 'base boundary') AND
        (sa.description IN ['directed angle occurrence',
            'closed profile occurrence'])
               )) = 1 )
             )) = 1;
WR6:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMTOIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
        (sa.name = 'target boundary') AND
        (sa.description IN ['directed angle occurrence',
            'closed profile occurrence'])
              )) = 1 )
             )) = 1;
WR7:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMTOIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
        sa.description = 'directed angle occurrence'
              )) <= 1 )
             )) <= 1;
WR8:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
        NOT ('AUTOMTOIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
              TYPEOF(pds)) OR
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
           'SHAPE_ASPECT.OF_SHAPE') |
        (sa.description = 'directed angle occurrence') AND
        (SIZEOF( QUERY( sdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
         'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
         ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
           TYPEOF(sdr)) AND
```

```
        (sdr.name = 'directed angle usage') AND
        ('AUTOMOTIVE_DESIGN.DIRECTED_ANGLE' IN
         TYPEOF(sdr.relating_shape_aspect))
              )) <> 1 )
             )) = 0 )
            )) = 0;
WR9:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
           'PROPERTY_DEFINITION.DEFINITION') |
       NOT ('AUTOMTOIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pds)) OR
       (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
       (sa.description = 'closed profile occurrence') AND
       (SIZEOF( QUERY( sdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
          TYPEOF(sdr)) AND
        (sdr.name = 'closed profile usage') AND
        (SIZEOF(['AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE',
         'AUTOMOTIVE_DESIGN.NGON_CLOSED_PROFILE',
         'AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE',
         'AUTOMOTIVE_DESIGN.CLOSED_PATH_PROFILE']
        * TYPEOF(sdr.relating_shape_aspect)) = 1 )
              )) <> 1 )
             )) = 0 )
            )) = 0;
WR10: SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
       (SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION' IN
         TYPEOF(pdr.used_representation))
              )) = 0 ) OR
       (SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION' IN
         TYPEOF(pdr.used_representation)) AND
        (pdr.used_representation.description = 'target face')
              )) = 1 )
            )) <= 1;
WR11: SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
           'PROPERTY_DEFINITION.DEFINITION') |
       ('AUTOMTOIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
         TYPEOF(pds)) AND
       (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
       (sa.name = 'target boundary') AND
       (sa.description = 'directed angle occurrence') AND
              (SIZEOF( QUERY( pdr <* USEDIN(sa, 'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       'AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION' IN
       TYPEOF(pdr.used_representation))) = 1 )
```

```
                )) = 1 )
               )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **feature_in_panel** shall have a **description** of either 'pocket', 'boss', or 'stairstep'.

WR2: The implicit representation of the **feature_in_panel** shall contain exactly three **representation_items** in its set of **items**.

WR3: The implicit representation of a **feature_in_panel** shall contain exactly one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'base fillet radius'.

WR4: The implicit representation of a **feature_in_panel** shall contain exaclty one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'target fillet radius'.

WR5: The **feature_in_panel** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'base boundary' and a **description** of either 'directed angle occurrence' or 'closed profile occurrence'.

WR6: The **feature_in_panel** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'target boundary' and a **description** of either 'directed angle occurrence' or 'closed profile occurrence'.

WR7: The **feature_in_panel** shall be the basis shape for at most one **shape_aspect** with a **description** of 'directed angle occurrence'.

WR8: Each **shape_aspect** of a **feature_in_panel** with a **description** of 'directed angle occurrence' shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'directed angle usage' in which the **relating_shape_aspect** is a **directed_angle.**

WR9: Each **shape_aspect** of a **feature_in_panel** with a **description** of 'closed profile occurrence' shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'closed profile usage' and a **relating_shape_aspect** that references a **circular_closed_profile, ngon_closed_profile, rectangular_closed_profile,** or **closed_path_profile.**

WR10: The **feature_in_panel** shall have at most one **connected_face_shape_representation.** If present, this **connected_face_shape_representation** shall have a **description** of 'target face'.

WR11: Each **feature_in_panel** that is the basis shape for a **shape_aspect** with a **name** of 'target boundary' and a **description** of 'directed angle occurrence' shall have a **connected_face_shape_representation.**

5.2.3.1.73 feature_pattern

A **feature_pattern** is a type of **replicate_feature** that relates a base feature and one or more **shape_aspects** which are the placement of the base feature at a specified location on the base part. See ARM definition for General_pattern in paragraph 4.2.204 for more information.

EXPRESS specification:

```
*)
ENTITY feature_pattern
  SUBTYPE OF (replicate_feature);
WHERE
  WR1:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
         (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
          ('AUTOMOTIVE_DESIGN.PLACEMENT' IN
            TYPEOF(srwp_i)) AND
          (srwp_i.name = 'added instance placement'))) > 0 )
              )) = 1
             )) = 1;
  WR2:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
         (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
          NOT ('AUTOMOTIVE_DESIGN.PLACEMENT' IN
                TYPEOF(srwp_i)) )) > 0 )
              )) > 0
             )) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The implicit representation of a **feature_pattern** shall contain one or more **representation_items** in its set of **items** of type **placement** with a name of 'added instance placement'.

WR2: The implicit representation of a **feature_pattern** shall only contain **representation_items** in its set of **items** that are of type **placement**.

5.2.3.1.74 featured_shape_representation

A **featured_shape_representation** is a type of **shape_representation** that represents the shape of a product that includes features. See ARM definition for Featured_shape in paragraph 4.2.179 for more information.

EXPRESS specification:

```
*)
ENTITY featured_shape_representation
  SUBTYPE OF (shape_representation);
WHERE
  WR1:  SIZEOF( QUERY( rr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'REPRESENTATION_RELATIONSHIP.REP_2') |
                 (rr.name = 'feature application sequence') AND
                 (SIZEOF( QUERY( pdr <* USEDIN(rr.rep_1, 'AUTOMOTIVE_DESIGN.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
                          'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
                           TYPEOF(pdr.definition)
                        )) = 1 )
              )) = 1;
  WR2:  SIZEOF( QUERY( pdr <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
                 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
                  TYPEOF(pdr.definition)
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: Each instance of a **featured_shape_representation** shall be the **rep_2** in a **representation_relationship** with a **name** of 'feature application sequence' which references as **rep_1** a **shape_representation** that represents a **product_definition_shape.**

WR2: The **featured_shape_representation** shall be the **shape_representation** of a **product_definition_shape.**

5.2.3.1.75 fillet

A **fillet** is a type of **transition_feature** that is the representation of a circular concave transition between two faces. See ARM definition for Fillet in paragraph 4.2.181 and for Constant_radius_fillet in paragraph 4.2.93 for more information.

EXPRESS specification:

```
*)
ENTITY fillet
```

```
  SUBTYPE OF (transition_feature);
WHERE
  WR1:  (SELF.description <> 'constant radius') XOR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)
               )) > 0 );
  WR2:  (SELF.description <> 'constant radius') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          ({1 <= SIZEOF(pdr.used_representation.items) <= 3})
               )) = 1 );
  WR3:  (SELF.description <> 'constant radius') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
           srwp_i.name IN ['radius','first face offset',
                   'second face offset']
                  )) = SIZEOF(pdr.used_representation.items))
               )) = 1 );
  WR4:  (SELF.description <> 'constant radius') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
            'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 ) AND
           (srwp_i.name = 'radius')
         )) = 1 )
               )) = 1 );
  WR5:  (SELF.description <> 'constant radius') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
            'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 ) AND
           (srwp_i.name = 'first face offset')
                 )) <= 1 )
               )) = 1 );
  WR6:  (SELF.description <> 'constant radius') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```

```
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                   (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
            'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 ) AND
           (srwp_i.name = 'second face offset')
         )) <= 1 )
              )) = 1 );
END_ENTITY;
(*
```

Formal propositions:

WR1: A **fillet** has an implicit representation if and only if it has a **description** of 'constant radius'.

WR2: If the **description** of the **fillet** is 'constant radius', the implicit representation shall contain between one and three **representation_items** in its set of **items**.

WR3: If the **description** of the **fillet** is 'constant radius', the implicit representation shall only contain **representation_items** in its set of **items** that have a **name** of either 'radius', 'first face offset', or 'second face offset'.

WR4: If the **description** of the **fillet** is 'constant radius', exactly one **representation_item** used for the implicit representation of the **fillet** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR5: If the **description** of the **fillet** is 'constant radius', at most one **representation_item** used for the implicit representation of the **fillet** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'first face offset'.

WR6: If the **description** of the **fillet** is 'constant radius', at most one **representation_item** used for the implicit representation of the **fillet** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'second face offset'.

5.2.3.1.76 group_member_assignment

A **group_member_assignment** is a type of **group_assignment** that assigns a group to a **group_member_item** that identifies a member of the group.

EXPRESS specification:

```
*)
ENTITY group_member_assignment
  SUBTYPE OF (group_assignment);
  items : SET [1:?] OF group_member_item;
WHERE
  WR1:  (SIZEOF( QUERY( i <* SELF.items |
          'AUTOMOTIVE_DESIGN.SHAPE_ASPECT' IN TYPEOF(i) )) = 0 ) OR
```

```
      (SELF.role.name = 'feature categorization');
WR2:  (SIZEOF( QUERY( i <* SELF.items |
        SIZEOF(['AUTOMOTIVE_DESIGN.DOCUMENT_TYPE',
        'AUTOMOTIVE_DESIGN.DRAWING_DEFINITION',
        'AUTOMOTIVE_DESIGN.APPROVAL_STATUS',
        'AUTOMOTIVE_DESIGN.SECURITY_CLASSIFICATION_LEVEL',
        'AUTOMOTIVE_DESIGN.CHARACTERIZED_PRODUCT_CATEGORY']
       * TYPEOF(i)) = 1
             )) = 0 ) OR
      ((SIZEOF(['AUTOMOTIVE_DESIGN.CLASSIFICATION_SYSTEM',
         'AUTOMOTIVE_DESIGN.EXTERNALLY_DEFINED_CLASSIFICATION_SYSTEM']
        * TYPEOF(SELF.assigned_group)) = 1 ) AND
       (SELF.role.name = 'classified member'));
WR3:  (SIZEOF( QUERY( i <* SELF.items |
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i) )) = 0 ) OR
      (SIZEOF( QUERY( i <* SELF.items |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)) AND
        ((i\product_definition.frame_of_reference.name <>
            'definitional template') OR
         ((SIZEOF( QUERY( pdr <* USEDIN( i, 'AUTOMOTIVE_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
           ('AUTOMOTIVE_DESIGN.ASSEMBLY_COMPONENT_RELATIONSHIP' IN
             TYPEOF(pdr)) AND
           (pdr.name = 'mating membership') )) +
           SIZEOF( QUERY( pdr <* USEDIN( i, 'AUTOMOTIVE_DESIGN.' +
            'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
           (pdr.name = 'collection membership') )) ) <> 1 ))
             )) = 0 );
WR4:  (SIZEOF( QUERY( i <* SELF.items |
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i) )) = 0 ) OR
      (SIZEOF( QUERY( i <* SELF.items |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)) AND
        (SIZEOF( QUERY( pdr <* USEDIN( i, 'AUTOMOTIVE_DESIGN.' +
         'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
         'AUTOMOTIVE_DESIGN.ASSEMBLY_COMPONENT_RELATIONSHIP' IN
          TYPEOF(pdr) )) = 0)))
       = SIZEOF( QUERY( i <* SELF.items |
           'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i) )) ) OR
      ((SIZEOF( QUERY( i <* SELF.items |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)) AND
        (SIZEOF( QUERY( pdr <* USEDIN( i, 'AUTOMOTIVE_DESIGN.' +
         'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
         'AUTOMOTIVE_DESIGN.ASSEMBLY_COMPONENT_RELATIONSHIP' IN
          TYPEOF(pdr) )) > 0)
       )) > 0 ) AND
       (SELF.assigned_group.name = 'assembly_definition') AND
       (SELF.role.name = 'assembly'));
WR5:  (SIZEOF( QUERY( i <* SELF.items |
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i) )) = 0) OR
      (SIZEOF( QUERY( i <* SELF.items |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)) AND
        (SIZEOF( QUERY( pdr <*  USEDIN( i, 'AUTOMOTIVE_DESIGN.' +
```

```
              'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
           pdr.name = 'mating membership')) = 0)))
         = SIZEOF( QUERY( i <* SELF.items |
             'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i) ))) OR
        ((SIZEOF( QUERY( i <* SELF.items |
          ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)) AND
          (SIZEOF( QUERY( pdr <* USEDIN( i, 'AUTOMOTIVE_DESIGN.' +
              'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
           pdr.name = 'mating membership' )) > 0 )
         )) > 0 ) AND
         (SELF.assigned_group.name = 'mating definition') AND
         (SELF.role.name = 'mating'));
  WR6:  (SIZEOF( QUERY( i <* SELF.items |
          'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i) )) = 0) OR
        (SIZEOF( QUERY( i <* SELF.items |
          ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)) AND
          (SIZEOF( QUERY( pdr <*  USEDIN( i, 'AUTOMOTIVE_DESIGN.' +
               'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
           pdr.name = 'collection membership')) = 0)))
         = SIZEOF( QUERY( i <* SELF.items |
             'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i) )))  OR
        ((SIZEOF( QUERY( i <* SELF.items |
          ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)) AND
          (SIZEOF( QUERY( pdr <* USEDIN( i, 'AUTOMOTIVE_DESIGN.' +
              'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
           pdr.name = 'collection membership' )) > 0 )
         )) > 0 ) AND
         (SELF.assigned_group.name = 'collection definition') AND
         (SELF.role.name = 'collection'));
  WR7:  (SIZEOF( QUERY( i <* SELF.items |
          'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_FEATURE_CATEGORY_USAGE' IN
          TYPEOF(i) )) = 0 ) OR
        ((SELF.assigned_group.name = 'influencing category group') AND
         (SELF.role.name = 'class usage group member'));
  WR8:  (SIZEOF( QUERY( i <* SELF.items |
          'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_FEATURE' IN
          TYPEOF(i) )) = 0 ) OR
        (('AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_FEATURE_CATEGORY' IN
           TYPEOF(SELF.assigned_group)) AND
         (SELF.role.name = 'specification category member'));
END_ENTITY;
(*
```

Attribute definitions:

items: The set of items that are members of the group.

Formal propositions:

WR1: If the **group_member_assignment** contains a **shape_aspect** in its set of **items**, it shall reference as its **role** a **group_role** that has a **name** of 'feature categorization'.

WR2: If the **group_member_assignment** contains a **document_type, drawing_definition, approval_status, security_classification_level, characterized_product_category,** or **externally_defined_product_category** in its set of **items**, it shall reference as its **assigned_group** either a **classification_system** or an **externally_defined_classification_system** and it shall reference as its **role** a **group_role** that has a **name** of 'classified member'.

WR3: If the **group_member_assignment** contains a **product_definition** in its set of **items**, this **product_definition** shall reference as its **frame_of_reference** a **product_definition_context** that has a **name** of 'definitional template' and shall be referenced as the **relating_product_definition** by either an **assembly_component_relationship,** by a **product_definition_relationship** that has a **name** of 'mating membership', or by a **product_definition_relationship** that has a **name** of 'collection membership' .

WR4: If the **group_member_assignment** contains a **product_definition** in its set of **items** that is referenced as the **relating_product_definition** by an **assembly_component_relationship,** it shall reference as its **assigned_group** a group that has a **name** of 'assembly definition' and shall reference as its **role** a **group_role** that has a **name** of 'assembly'.

WR5: If the **group_member_assignment** contains a **product_definition** in its set of **items** that is referenced as the **relating_product_definition** by a **product_definition_relationship** that has a **name** of 'mating membership', it shall reference as its **assigned_group** a group that has a **name** of 'mating definition' and shall reference as its **role** a **group_role** that has a **name** of 'mating'.

WR6: If the **group_member_assignment** contains a **product_definition** in its set of **items** that is referenced as the **relating_product_definition** by a **product_definition_relationship** that has a **name** of 'collection membership', it shall reference as its **assigned_group** a group that has a **name** of 'collection definition' and shall reference as its **role** a **group_role** that has a **name** of 'collection'.

WR7: If the **group_member_assignment** contains a **product_concept_feature_category_usage** in its set of **items**, it shall reference as its **assigned_group** a **group** that has a **name** of 'influencing category group' and shall reference as its **role** a **group_role** that has a **name** of 'class usage group member'.

WR8: If the **group_member_assignment** contains a **product_concept_feature** in its set of **items**, it shall reference as its **assigned_group** a **product_concept_feature_category** and shall reference as its **role** a **group_role** that has a **name** of 'specification category member'.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **group_member_assignment** entity:

– restrict_classification_system_for_approval_status (see 5.2.4.76);

– restrict_classification_system_for_document_type (see 5.2.4.77);

– restrict_classification_system_for_security_classification_level (see 5.2.4.78).

5.2.3.1.77 hidden_element_over_riding_styled_item

The **hidden_element_over_riding_styled_item** is a type of **context_dependent_over_riding_-styled_item** that specifies the handling of hidden elements. See ARM definition for Hidden_-element_handling in paragraph 4.2.227 for more information.

EXPRESS specification:

```
*)
ENTITY hidden_element_over_riding_styled_item
  SUBTYPE OF (context_dependent_over_riding_styled_item);
WHERE
  WR1:  SIZEOF(SELF.style_context) = 1;
  WR2:  'AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN TYPEOF
                (SELF.style_context[1]);
  WR3:  'AUTOMOTIVE_DESIGN.CAMERA_IMAGE' IN TYPEOF
                (SELF.item);
  WR4:  'AUTOMOTIVE_DESIGN.CAMERA_MODEL_D3_WITH_HLHSR' IN TYPEOF
                (SELF.item\mapped_item.mapping_source.mapping_origin);
  WR5:  SIZEOF( QUERY( rep <* USEDIN(SELF,
                 'AUTOMOTIVE_DESIGN.REPRESENTATION.ITEMS') |
                    'AUTOMOTIVE_DESIGN.PRESENTATION_VIEW'
                    IN TYPEOF(rep)
        )) > 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: There shall be exactly one **style_context.**

WR2: The **style_context** shall be a **presentation_view.**

WR3: The item where the overriding style is applied for all hidden elements shall be a **camera_-image.**

WR4: The **mapping_origin** of the item shall be a **camera_model_d3_with_hlhsr.**

WR5: The **hidden_element_over_riding_styled_item** shall be included in the set of items of a **presentation_view.**

5.2.3.1.78 hole_bottom

A **hole_bottom** is a type of **shape_aspect** that is the end condition for a **round_hole.** See ARM definition for Conical_bottom in paragraph 4.2.90, for Flat_bottom in paragraph 4.2.183, for Flat_with_chamfer_bottom in paragraph 4.2.185, for Flat_with_radius_bottom in paragraph

4.2.186, for Spherical_bottom in paragraph 4.2.421, and for Through_hole_condition in paragraph 4.2.460 for more information.

EXPRESS specification:

```
*)
ENTITY hole_bottom
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SELF.description IN ['through','flat','flat with chamfer',
              'flat with radius','spherical','conical'];
  WR3:  NOT (SELF.description IN ['through','flat with chamfer']) OR
        (SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 0);
  WR4:  NOT (SELF.description IN ['flat','flat with radius','spherical','conical']) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)
               )) = 1 );
  WR5:  (SELF.description <> 'flat') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
          (SIZEOF(pdr.used_representation.items) = 1)
               )) = 1 );
  WR6:  NOT (SELF.description IN ['flat with radius','spherical']) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
          (SIZEOF(pdr.used_representation.items) = 2)
               )) = 1 );
  WR7:  (SELF.description <> 'conical') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
          (SIZEOF(pdr.used_representation.items) = 3)
               )) = 1 );
  WR8:  NOT (SELF.description IN ['flat','flat with radius','spherical','conical']) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
           (srwp_i.name = 'bottom location') AND
               (srwp_i.description IN ['part start','part end']) AND
```

```
          ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
            TYPEOF(srwp_i))
       )) = 1)
             )) = 1 );
WR9:  (SELF.description <> 'flat with radius') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
         (srwp_i.name = 'corner radius') AND
         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
          'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
         * TYPEOF(srwp_i)) = 2 )
       )) = 1)
             )) = 1 );
WR10: (SELF.description <> 'spherical') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
         (srwp_i.name = 'radius') AND
         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
          'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
         * TYPEOF(srwp_i)) = 2 )
       )) = 1)
             )) = 1 );
WR11: (SELF.description <> 'conical') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
         (srwp_i.name = 'tip radius') AND
         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
          'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
         * TYPEOF(srwp_i)) = 2 )
       )) = 1)
             )) = 1 );
WR12: (SELF.description <> 'conical') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
         (srwp_i.name = 'tip angle') AND
         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
          'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
         * TYPEOF(srwp_i)) = 2 )
       )) = 1)
             )) = 1 );
```

```
  WR13: (SELF.description <> 'flat with chamfer') OR
        (SIZEOF( QUERY( fcr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
               'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(fcr)) AND
          (fcr.name = 'first chamfer offset') AND
          ('AUTOMOTIVE_DESIGN.CHAMFER_OFFSET' IN TYPEOF(fcr.related_shape_aspect))
               )) = 1 );
  WR14: (SELF.description <> 'flat with chamfer') OR
        (SIZEOF( QUERY( fcr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
               'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(fcr)) AND
          (fcr.name = 'second chamfer offset') AND
          ('AUTOMOTIVE_DESIGN.CHAMFER_OFFSET' IN TYPEOF(fcr.related_shape_aspect))
               )) = 1 );
  WR15: (SIZEOF( QUERY( fcr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(fcr)) AND
          (fcr.description = 'hole bottom usage') AND
          ('AUTOMOTIVE_DESIGN.ROUND_HOLE' IN
            TYPEOF(fcr.related_shape_aspect.of_shape.definition)) AND
          (fcr.related_shape_aspect.of_shape.definition.description =
            'bottom condition occurrence')
               )) >= 1 );
END_ENTITY;
(*
```

Formal propositions:

WR1: The **hole_bottom** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **description** of the **hole_bottom** shall be either 'through', 'flat', 'flat with chamfer', 'flat with radius', 'spherical', or 'conical'.

WR3: If the **description** of the **hole_bottom** is 'flat with chamfer' or 'through', the **hole_bottom** shall not have any representation.

WR4: If the **description** of the **hole_bottom** is 'flat', 'flat with radius', 'spherical', or 'conical', the **hole_bottom** shall have exactly one **shape_representation_with_parameters** to specify its implicit representation.

WR5: If the **description** of the **hole_bottom** is 'flat', the **shape_representation_with_parameters** that represents the **hole_bottom** shall contain exactly one **representation_item** in its set of **items**.

WR6: If the **description** of the **hole_bottom** is 'flat with radius' or 'spherical', the **shape_representation_with_parameters** that represents the **hole_bottom** shall contain exactly two **representation_items** in its set of **items**.

WR7: If the **description** of the **hole_bottom** is 'conical', the **shape_representation_with_parameters** that represents the **hole_bottom** shall contain exactly three **representation_items** in its set of **items**.

WR8: If the **description** of the **hole_bottom** is 'flat', 'flat with radius', 'spherical', or 'conical', exactly one **representation_item** used for the implicit representation of a **hole_bottom** shall be of type **descriptive_representation_item** with a **name** of 'bottom location' and a **description** of either 'path start' or 'path end'.

WR9: If the **description** of the **hole_bottom** is 'flat with radius', exactly one **representation_item** used for the implicit representation of a **hole_bottom** shall have a **name** of 'corner radius'. This **representation_item** shall be of type **measure_representation_item** and **length_measure_with_unit.**

WR10: If the **description** of the **hole_bottom** is 'spherical', exactly one **representation_item** used for the implicit representation of a **hole_bottom** shall have a **name** of 'radius'. This **representation_item** shall be of type **measure_representation_item** and **length_measure_with_unit.**

WR11: If the **description** of the **hole_bottom** is 'conical', exactly one **representation_item** used for the implicit representation of a **hole_bottom** shall have a **name** of 'tip radius'. This **representation_item** shall be of type **measure_representation_item** and **length_measure_with_unit.**

WR12: If the **description** of the **hole_bottom** is 'conical', exactly one **representation_item** used for the implicit representation of a **hole_bottom** shall have a **name** of 'tip angle'. This **representation_item** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit.**

WR13: If the **description** of the **hole_bottom** is 'flat with chamfer', the **hole_bottom** shall be the **relating_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'first chamfer offset' in which the **related_shape_aspect** is a **chamfer_offset.**

WR14: If the **description** of the **hole_bottom** is 'flat with chamfer', the **hole_bottom** shall be the **relating_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'second chamfer offset' in which the **related_shape_aspect** is a **chamfer_offset.**

WR15: The **hole_bottom** shall be the **relating_shape_aspect** in at least one **feature_component_relationship** with a **description** of 'hole bottom usage' in which the **related_shape_aspect** is an aspect of the shape of a **round_hole** with a **description** of 'bottom condition occurrence'.

5.2.3.1.79 hole_in_panel

A **hole_in_panel** is a type of **feature_definition** applicable to panel parts. A **hole_in_panel** is an opening in a panel bounded by a closed profile that is created by a stamping process. A **hole_in_panel** may have a barring if the punched material is partially bended in the punch direction. See ARM definition for Hole_feature_in_panel in paragraph 4.2.229 and for Panel_hole_with_barring in paragraph 4.2.299 for more information.

EXPRESS specification:

```
*)
ENTITY hole_in_panel
  SUBTYPE OF (feature_definition);
WHERE
  WR1:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pds)) AND
         (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
            'SHAPE_ASPECT.OF_SHAPE') |
         (sa.name = 'hole profile') AND
         (sa.description = 'closed profile occurrence') AND
         (SIZEOF( QUERY( sdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
            TYPEOF(sdr)) AND
          (sdr.name = 'closed profile usage') AND
          (SIZEOF(['AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE',
           'AUTOMOTIVE_DESIGN.NGON_CLOSED_PROFILE',
           'AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE',
           'AUTOMOTIVE_DESIGN.CLOSED_PATH_PROFILE']
          * TYPEOF(sdr.relating_shape_aspect)) = 1 )
                )) = 1 )
                )) = 1 )
              )) = 1;
  WR2:  (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pds)) AND
         (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT.OF_SHAPE') |
          (sa.name = 'taper') AND
          (sa.description = 'taper occurrence')
                )) = 0 )
               )) = 0 ) OR
        (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pds)) AND
         (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
           'SHAPE_ASPECT.OF_SHAPE') |
         (sa.name = 'taper') AND
         (sa.description = 'taper occurrence') AND
         (SIZEOF( QUERY( fcr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
               'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(fcr)) AND
          (fcr.name = 'taper usage') AND
          ('AUTOMOTIVE_DESIGN.TAPER' IN
           TYPEOF(fcr.relating_shape_aspect))
```

```
                )) = 1 )
                )) = 1 )
              )) = 1 );
  WR3:  (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pds)) AND
         (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
            'SHAPE_ASPECT.OF_SHAPE') |
          (sa.name = 'barring profile') AND
          (sa.description = 'closed profile occurrence')
                )) = 0 )
               )) = 0 ) OR
        (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pds)) AND
         (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
            'SHAPE_ASPECT.OF_SHAPE') |
          (sa.name = 'barring profile') AND
          (sa.description = 'closed profile occurrence') AND
          (SIZEOF( QUERY( sdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
         'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
           ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
             TYPEOF(sdr)) AND
           (sdr.name = 'closed profile usage')
         )) = 1 )
         )) = 1 )
               )) = 1 );
END_ENTITY;
(*
```

Formal propositions:

WR1: The **hole_in_panel** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'hole profile' and a **description** of 'closed profile occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'closed profile usage' and a **relating_shape_aspect** that references a **circular_closed_profile**, **ngon_closed_profile**, **rectangular_closed_profile**, or **closed_path_profile.**

WR2: The **hole_in_panel** shall be the basis shape for at most one **shape_aspect** with a **name** of 'taper' and a **description** of 'taper occurrence'. If present, this **shape_aspect** shall be the **related_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'taper usage' in which the **relating_shape_aspect** is a **taper**.

WR3: The **hole_in_panel** shall be the basis shape for at most one **shape_aspect** with a **name** of 'barring profile' and a **description** of 'closed profile occurrence'. If present, this **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'closed profile usage'.

Informal propositions:

IP1: The barring profile shall lie in the same plane as the hole profile.

IP2: The barring profile shall lie completely inside the hole profile.

5.2.3.1.80 hybrid_3d_shape_representation

A **hybrid_shape_representation** is a type of **shape_representation** that represents the shape or a portion of the shape of a product using different types of **shape_representations.** Both geometrically bounded and topologically bounded **shape_representations** are allowed. See ARM definition for Hybrid_geometric_model_3d in paragraph 4.2.230 for more information.

EXPRESS specification:

```
*)
ENTITY hybrid_3d_shape_representation
  SUBTYPE OF (shape_representation);
WHERE
  WR1:  SIZEOF (QUERY (i <* SELF\representation.items |
                NOT (SIZEOF (['AUTOMOTIVE_DESIGN.' +
                              'GEOMETRIC_REPRESENTATION_ITEM',
                              'AUTOMOTIVE_DESIGN.MAPPED_ITEM'] *
                            TYPEOF(i)) = 1))) = 0;
  WR2:  SIZEOF (QUERY (mi <* QUERY (item <* SELF\representation.items |
                       'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(item)) |
                       NOT ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN
                       TYPEOF(mi\mapped_item.mapping_source.mapped_representation))
               )) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: A **hybrid_3d_shape_representation** shall contain **representation_items** that are of type **geometric_representation_item** or **mapped_item.**

WR2: If there is a **mapped_item** in a **hybrid_3d_shape_representation,** the source of the **mapped_item** shall be a **shape_representation.**

5.2.3.1.81 instanced_feature

An **instanced_feature** is a type of **shape_aspect** and of **feature_definition** that is the identification of a preconceived form pattern on a part. See ARM definition for Included_feature in paragraph 4.2.232 for more information.

EXPRESS specification:

```
*)
ENTITY instanced_feature
  SUBTYPE OF (feature_definition, shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
END_ENTITY;
(*
```

Formal propositions:

WR1: The **instanced_feature** shall be an aspect of the shape of a **product_definition.**

5.2.3.1.82 joggle

A **joggle** is a type of **feature_definition** applicable to panel parts. A **joggle** is a change in elevation of a face along a rim desgned to accommodate either clearance for the insertion of another part or a contact point for attachment of another part. See ARM definition for Joggle_feature in paragraph 4.2.250 for more information.

EXPRESS specification:

```
*)
ENTITY joggle
  SUBTYPE OF (feature_definition);
WHERE
  WR1:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
         (SIZEOF(pdr.used_representation.items) <= 6 )
               )) = 1
              )) = 1;
  WR2:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
         (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
          (SIZEOF(['AUTOMOTIVE_DESIGN.' +
           'MEASURE_REPRESENTATION_ITEM',
           'AUTOMOTIVE_DESIGN.' +
           'PLANE_ANGLE_MEASURE_WITH_UNIT']
          * TYPEOF(srwp_i)) = 2) AND
```

```
        (srwp_i.name = 'direction angle at path end')
              )) = 1 )
             )) = 1
            )) = 1;
WR3:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
       SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
       (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
        (SIZEOF(['AUTOMOTIVE_DESIGN.' +
         'MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.' +
         'PLANE_ANGLE_MEASURE_WITH_UNIT']
        * TYPEOF(srwp_i)) = 2) AND
        (srwp_i.name = 'direction angle at path start')
              )) = 1 )
             )) = 1
            )) = 1;
WR4:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
       SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
       (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
        (SIZEOF(['AUTOMOTIVE_DESIGN.' +
         'MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.' +
         'LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(srwp_i)) = 2) AND
        (srwp_i.name = 'height')
              )) = 1 )
             )) = 1
            )) = 1;
WR5:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
       SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
       (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
        (SIZEOF(['AUTOMOTIVE_DESIGN.' +
         'MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.' +
         'PLANE_ANGLE_MEASURE_WITH_UNIT']
        * TYPEOF(srwp_i)) = 2) AND
        (srwp_i.name = 'slope angle at path end')
              )) = 1 )
             )) = 1
            )) = 1;
```

```
WR6:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
       SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
       (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
        (SIZEOF(['AUTOMOTIVE_DESIGN.' +
         'MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.' +
         'PLANE_ANGLE_MEASURE_WITH_UNIT']
        * TYPEOF(srwp_i)) = 2) AND
        (srwp_i.name = 'slope angle at path start')
              )) = 1 )
             )) = 1
            )) = 1;
WR7:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
         TYPEOF(pds)) AND
       (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
       (sa.name = 'defining path') AND
       (sa.description = 'path feature component occurrence') AND
       (SIZEOF( QUERY( sdr <* USEDIN(sa, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
          TYPEOF(sdr)) AND
        (sdr.name = 'path feature component usage') AND
        ('AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN
         TYPEOF(sdr.relating_shape_aspect)) AND
        (sdr.relating_shape_aspect.description = 'linear')
              )) = 1 )
              )) = 1 )
            )) = 1;
WR8:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
         TYPEOF(pds)) AND
       (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
       (sa.name = 'termination') AND
       (sa.description = 'joggle termination occurrence') AND
       (SIZEOF( QUERY( sdr <* USEDIN(sa, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
          TYPEOF(sdr)) AND
        (sdr.name = 'joggle termination usage') AND
        ('AUTOMOTIVE_DESIGN.JOGGLE_TERMINATION' IN
         TYPEOF(sdr.relating_shape_aspect))
              )) = 1 )
              )) = 1 )
```

```
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The implicit representation of a **joggle** shall contain six **representation_items** in its set of **items**.

WR2: The implicit representation of a **joggle** shall contain exactly one **representation_item** of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'direction angle at path end'.

WR3: The implicit representation of a **joggle** shall contain exactly one **representation_item** of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'direction angle at path start'.

WR4: The implicit representation of a **joggle** shall contain exactly one **representation_-item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'height'.

WR5: The implicit representation of a **joggle** shall contain exactly one **representation_item** of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'slope angle at path end'.

WR6: The implicit representation of a **joggle** shall contain exactly one **representation_item** of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'slope angle at path start'.

WR7: The **joggle** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'defining path' and a **description** of 'path feature component occurrence' that is the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'path feature component usage' in which the **relating_shape_aspect** is a **path_feature_component** with a **description** of 'linear'.

WR8: The **joggle** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'termination' and a **description** of 'joggle termination occurrence' that is the **related_shape_-aspect** in exactly one **feature_component_relationship** with a **name** of 'joggle termination usage' in which the **relating_shape_aspect** is a **joggle_termination.**

5.2.3.1.83 joggle_termination

A **joggle_termination** is a type of shape aspect that is the represenation of the end condition of a **joggle**. See ARM definition for Joggle_termination in paragraph 4.2.251, for Face_termination in paragraph 4.2.169, and for Value_termination in paragraph 4.2.484 for more information.

EXPRESS specification:

```
*)
```

```
ENTITY joggle_termination
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SELF.description IN ['value termination','face termination'];
  WR3:  (SELF.description <> 'value termination') XOR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)
               )) > 0 );
  WR4:  (SELF.description <> 'value termination') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
          (SIZEOF(pdr.used_representation.items) = 2)
               )) > 0 );
  WR5:  (SELF.description <> 'value termination') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
           (srwp_i.name = 'depth') AND
           (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
            'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)
          ) = 2 )
         )) = 1 )
               )) = 1 );
  WR6:  (SELF.description <> 'value termination') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
           (srwp_i.name = 'angle') AND
           (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
            'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)
          ) = 2 )
         )) = 1 )
               )) = 1 );
  WR7:  SIZEOF( QUERY( fcs <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
               'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
         ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
           TYPEOF(fcs)) AND
         (fcs.name = 'joggle termination usage') AND
         ('AUTOMOTIVE_DESIGN.JOGGLE' IN
          TYPEOF(fcs.related_shape_aspect.of_shape.definition)) AND
```

```
        (fcs.related_shape_aspect.name = 'termination') AND
        (fcs.related_shape_aspect.description = 'joggle termination occurrence')
            )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **joggle_termination** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **description** for the **joggle_termination** shall be either 'value termination' or 'face termination'.

WR3: A **joggle_termination** has an implicit representation if and only if it has a **description** of 'value termination'.

WR4: If the **description** of the **joggle_termination** is 'value termination', its implicit representation shall contain exactly two **representation_items** in its set of **items**.

WR5: If the **description** of the **joggle_termination** is 'value termination', exactly one **representation_item** used for the implicit representation of the **joggle_termination** shall be of type **measure_representation_item** and **length_measure_with_unit** with a name of 'depth'.

WR6: If the **description** of the **joggle_termination** is 'value termination', exactly one **representation_item** used for the implicit representation of the **joggle_termination** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a name of 'angle'.

WR7: The **joggle_termination** shall be the **relating_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'joggle termination usage' in which the **related_shape_aspect** is a **shape_aspect** of a **joggle** with a **name** of 'termination' and a **description** of 'joggle termination occurrence'.

5.2.3.1.84 known_source

A **known_source** is a type of **external_source** and of **pre_defined_item** that provides a mechanism to refer to an entry in an external library due to ISO 13584. See ARM definition for External_library_reference in paragraph 4.2.165 for more information.

EXPRESS specification:

```
*)
ENTITY known_source
  SUBTYPE OF (external_source, pre_defined_item);
WHERE
  WR1:  SELF.name = 'ISO 13584 dictionary';
END_ENTITY;
(*
```

Formal propositions:

WR1: The **known_source** shall have a **name** of 'ISO 13584 dictionary'.

5.2.3.1.85 location_shape_representation

A **location_shape_representation** is a type of **representation** that represents a geometric position or a reference point.

EXPRESS specification:

```
*)
ENTITY location_shape_representation
  SUBTYPE OF (representation);
WHERE
  WR1:  SIZEOF(SELF.items) = 1;
  WR2:  SIZEOF( QUERY( i <* SELF.items|
                'AUTOMOTIVE_DESIGN.POINT' IN TYPEOF(i)
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **location_shape_representation** shall have exactly one **representation_item** in its set of **items**.

WR2: The geometric element that is used to represent the **location_shape_representation** shall be a **point**.

5.2.3.1.86 locator

A **locator** is a type of **feature_definition** applicable to panel parts. A **locator** is a conical impression used to locate the panel in the tooling. See ARM definition for Locator_feature in paragraph 4.2.271 for more information.

EXPRESS specification:

```
*)
ENTITY locator
  SUBTYPE OF (feature_definition);
WHERE
  WR1:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
              SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                      SIZEOF(pdr.used_representation.items) = 4
                    )) = 1
```

```
            )) = 1;
  WR2:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
           'PROPERTY_DEFINITION.DEFINITION') |
            SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                   ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
              TYPEOF(pdr.used_representation)) AND
                   (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
             (srwp_i.name = 'diameter') AND
             (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                              'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(srwp_i)
             ) = 2 )
            )) = 1 )
                  )) = 1
            )) = 1;
  WR3:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
           'PROPERTY_DEFINITION.DEFINITION') |
            SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                   ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
              TYPEOF(pdr.used_representation)) AND
                   (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
             (srwp_i.name = 'tip radius') AND
             (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                              'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(srwp_i)
             ) = 2 )
            )) = 1 )
                  )) = 1
            )) = 1;
  WR4:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
           'PROPERTY_DEFINITION.DEFINITION') |
            SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                   ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
              TYPEOF(pdr.used_representation)) AND
                   (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
             (srwp_i.name = 'tip angle') AND
             (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                              'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
        TYPEOF(srwp_i)
             ) = 2 )
            )) = 1 )
                  )) = 1
            )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The implicit representation of the locator shall contain four **representation_items** in its set of **items**.

WR2: The implicit representation of the **locator** shall contain exactly one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'diameter'.

WR3: The implicit representation of the **locator** shall contain exactly one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'tip radius'.

WR4: The implicit representation of the **locator** shall contain exactly one **representation_item** of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'tip angle'.

5.2.3.1.87 mated_item_relationship

A **mated_item_relationship** is a type of **product_definition** and of **product_definition_relationship** that specifies the physical connection of two **product_definition_occurrence** instances within a mated product independent from the assembly structure of this product. See ARM definition for Mated_item_association in paragraph 4.2.275 for more information.

EXPRESS specification:

```
*)
ENTITY mated_item_relationship
  SUBTYPE OF (product_definition, product_definition_relationship);
WHERE
  WR1:  (SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
         'PRODUCT_DEFINITION_RELATIONSHIP.' +
         'RELATED_PRODUCT_DEFINITION')) = 1) AND
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PRODUCT_DEFINITION_RELATIONSHIP.' +
                'RELATED_PRODUCT_DEFINITION') |
                 (pdr.name = 'mating context') AND
                 (SIZEOF( QUERY( gma <* USEDIN(pdr.relating_product_definition,
                                 'AUTOMOTIVE_DESIGN.GROUP_MEMBER_ASSIGNMENT.ITEMS') |
                          (gma.assigned_group.name = 'mating definition') AND
                          (gma.role.name = 'mating')
                        )) = 1 )
               )) = 1);
  WR2:  ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_OCCURRENCE' IN
         TYPEOF(SELF.relating_product_definition)) AND
        (SIZEOF( QUERY( pdr <* USEDIN(SELF.relating_product_definition,
                        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RELATIONSHIP.' +
                        'RELATED_PRODUCT_DEFINITION') |
                 (pdr.name = 'mating membership') AND
                 (SIZEOF( QUERY( pdr1 <* USEDIN(pdr.relating_product_definition,
```

```
                                  'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RELATIONSHIP.' +
                                  'RELATING_PRODUCT_DEFINITION') |
                          (pdr1.name = 'mating context') AND
                          (pdr1.related_product_definition :=: SELF)
                        )) = 1 )
                )) = 1);
  WR3:  ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_OCCURRENCE' IN
         TYPEOF(SELF.related_product_definition)) AND
        (SIZEOF( QUERY( pdr <* USEDIN(SELF.related_product_definition,
                        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RELATIONSHIP.' +
                        'RELATED_PRODUCT_DEFINITION') |
                  (pdr.name = 'mating membership') AND
                  (SIZEOF( QUERY( pdr1 <* USEDIN(pdr.relating_product_definition,
                                  'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RELATIONSHIP.' +
                                  'RELATING_PRODUCT_DEFINITION') |
                          (pdr1.name = 'mating context') AND
                          (pdr1.related_product_definition :=: SELF)
                        )) = 1 )
                )) = 1);
END_ENTITY;
(*
```

Formal propositions:

WR1: The **mated_item_relationship** shall be referenced as the **related_product_definition** in exactly one **product_definition_relationship**. This **product_definition_relationship** shall have a **name** of 'mating context' and shall reference as the **relating_product_definition** a **product_definition** that is contained in the set of **items** of a **group_member_assignment** that references as its **assigned_group** a group that has a **name** of 'mating definition' and that references as its **role** a **group_role** that has a **name** of 'mating'

WR2: The **mated_item_relationship** shall reference as **relating_product_definition** a **product_definition_occurrence** that is referenced as the **related_product_definition** in exactly one **product_definition_relationship** that has a **name** of 'mating membership' and that references as the **relating_product_definition** the **product_definition** that is referenced as the **relating_product_definition** by the **product_definition_relationship** with a **name** of 'mating context' in which the **mated_item_relationship** is the **related_product_definition.**

WR3: The **mated_item_relationship** shall reference as **related_product_definition** a **product_definition_occurrence** that is referenced as the **related_product_definition** in exactly one **product_definition_relationship** that has a **name** of 'mating membership' and that references as the **relating_product_definition** the **product_definition** that is referenced as the **relating_product_definition** by the **product_definition_relationship** with a **name** of 'mating context' in which the **mated_item_relationship** is the **related_product_definition.**

5.2.3.1.88 model_change

A **model_change** is a type of **property_definition** that describes the differences between two objects concerning the properties of these objects. See ARM definition for Model_change in

paragraph 4.2.283 for more information.

EXPRESS specification:

```
*)
ENTITY model_change
  SUBTYPE OF (property_definition);
WHERE
  WR1:  SIZEOF(['AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RELATIONSHIP',
                'AUTOMOTIVE_DESIGN.SHAPE_ASPECT_RELATIONSHIP']
               * TYPEOF(SELF.definition)) = 1;
  WR2:  (SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) <= 4) AND
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                 NOT (pdr.name IN ['added elements','deleted elements',
                                   'relating change description',
                                   'related change description'])
               )) = 0);
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                pdr.name = 'added elements'
               )) <= 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                pdr.name = 'deleted elements'
               )) <= 1;
  WR5:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                pdr.name = 'relating change description'
               )) = 1;
  WR6:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                pdr.name = 'related change description'
               )) = 1;
  WR7:  (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                 (pdr.name = 'relating change description')
               )) = 0 ) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            (pdr.name = 'relating change description') AND
            (SIZEOF( QUERY( rr <* USEDIN(pdr.used_representation,
                        'AUTOMOTIVE_DESIGN.REPRESENTATION_RELATIONSHIP.REP_1') |
                    (SIZEOF( QUERY( pdr_1 <* USEDIN(rr.rep_2,
                                'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.' +
                                'USED_REPRESENTATION') |
                            pdr_1.name = 'related change description'
                           )) = 1) AND
                    (rr.name = 'change description')
```

```
                    )) = 1 )
                )) = 1 );
END_ENTITY;
(*
```

Formal propositions:

WR1: The **model_change** shall reference as its **definition** either a **product_definition_relationship** or a **shape_aspect_relationship.**

WR2: Each **model_change** shall be referenced by at most four **property_definition_representation**s. The **name** of this **property_definition_representation**s shall be either 'added elements','deleted elements', 'relating change description', or 'related change description'.

WR3: There shall be at most one **property_definition_representation** with a **name** of 'added elements' referencing a **model_change.**

WR4: There shall be at most one **property_definition_representation** with a **name** of 'deleted elements' referencing a **model_change.**

WR5: There shall be at exactly one **property_definition_representation** with a **name** of 'relating change description' referencing a **model_change.**

WR6: There shall be at exactly one **property_definition_representation** with a **name** of 'related change description' referencing a **model_change.**

WR7: The representation, that is referenced as **used_representation** by the **property_definition_representation** with a **name** of 'relating change description' shall play the role of **rep_1** in a **representation_relationship** where the role of **rep_2** is played by a representation that is referenced as **used_representation** by the **property_definition_representation** with a **name** of 'related change description'. This **representation_relationship** shall have a **name** of 'change description'.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **model_change** entity:

– restrict_multi_language_for_model_change (see 5.2.4.122).

5.2.3.1.89 modified_pattern

A **modified_pattern** is a type of **replicate_feature** that represents the result of a collection of omissions and offsets of a base feature in either a **rectangular_pattern** or a **circular_pattern.** See ARM definition for Circular_offset_pattern in paragraph 4.2.58, for Circular_omit_pattern in paragraph 4.2.59, for Rectangular_offset_pattern in paragraph 4.2.371, and for Rectangular_omit_pattern in paragraph 4.2.372 for more information.

EXPRESS specification:

```
*)
ENTITY modified_pattern
```

```
  SUBTYPE OF (replicate_feature);
WHERE
  WR1:  SIZEOF( QUERY( cor <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'CHARACTERIZED_OBJECT_RELATIONSHIP.RELATED_OBJECT') |
         (cor.name = 'replicate base assignment') AND
         (SIZEOF(['AUTOMOTIVE_DESIGN.CIRCULAR_PATTERN',
          'AUTOMOTIVE_DESIGN.RECTANGULAR_PATTERN']
         * TYPEOF(cor.relating_object)) = 1)
             )) = 1;
  WR2:  (SIZEOF( QUERY( cor <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'CHARACTERIZED_OBJECT_RELATIONSHIP.RELATED_OBJECT') |
          (cor.name = 'replicate base assignment') AND
          ('AUTOMOTIVE_DESIGN.CIRCULAR_PATTERN' IN
            TYPEOF(cor.relating_object))
               )) = 0 ) OR
        (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pds)) AND
          (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
                'SHAPE_ASPECT.OF_SHAPE') |
           NOT ( sa.description IN ['circular omit','circular offset'])
         )) > 0 )
               )) = 0);
  WR3:  (SIZEOF( QUERY( cor <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'CHARACTERIZED_OBJECT_RELATIONSHIP.RELATED_OBJECT') |
          (cor.name = 'replicate base assignment') AND
          ('AUTOMOTIVE_DESIGN.CIRCULAR_PATTERN' IN
            TYPEOF(cor.relating_object))
               )) = 0 ) OR
        (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pds)) AND
          (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
                'SHAPE_ASPECT.OF_SHAPE') |
           sa.description IN ['circular omit','circular offset']
         )) > 0 )
               )) >= 1);
  WR4:  (SIZEOF( QUERY( cor <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'CHARACTERIZED_OBJECT_RELATIONSHIP.RELATED_OBJECT') |
          (cor.name = 'replicate base assignment') AND
          ('AUTOMOTIVE_DESIGN.RECTANGULAR_PATTERN' IN
            TYPEOF(cor.relating_object))
               )) = 0 ) OR
        (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pds)) AND
          (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
                'SHAPE_ASPECT.OF_SHAPE') |
           NOT ( sa.description IN ['rectangular omit','rectangular offset'])
```

```
       )) > 0 )
             )) = 0);
WR5:  (SIZEOF( QUERY( cor <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'CHARACTERIZED_OBJECT_RELATIONSHIP.RELATED_OBJECT') |
         (cor.name = 'replicate base assignment') AND
         ('AUTOMOTIVE_DESIGN.RECTANGULAR_PATTERN' IN
           TYPEOF(cor.relating_object))
             )) = 0 ) OR
      (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pds)) AND
         (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT.OF_SHAPE') |
          sa.description IN ['rectangular omit','rectangular offset']
       )) > 0 )
             )) >= 1);
WR6:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT.OF_SHAPE') |
         SIZEOF( QUERY( pdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation))
              )) = 1
              )) = SIZEOF(USEDIN( pds, 'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE')))
            )) = SIZEOF( QUERY(pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
           'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pd) ));
WR7:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT.OF_SHAPE') |
         (sa.description <> 'circular omit') OR
         (SIZEOF( QUERY( pdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( pdr.items) = 1)
        )) = 1)
              )) = SIZEOF(USEDIN( pds, 'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE')))
            )) = SIZEOF( QUERY(pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
           'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pd) ));
WR8:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
```

```
              'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT.OF_SHAPE') |
         (sa.description <> 'circular omit') OR
         (SIZEOF( QUERY( pdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
          ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN
            TYPEOF(r_i)) AND
          ('AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN
          TYPEOF(r_i\measure_representation_item.value_component)) AND
          (r_i.name = 'omit index') )) = 1 )
        )) = 1)
        )) = SIZEOF(USEDIN( pds, 'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE')))
             )) = SIZEOF( QUERY(pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
           'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pd) ));
WR9:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT.OF_SHAPE') |
         NOT (sa.description IN ['circular offset','rectangular omit']) OR
         (SIZEOF( QUERY( pdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( pdr.items) = 2)
        )) = 1)
               )) = SIZEOF(USEDIN( pds, 'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE')))
             )) = SIZEOF( QUERY(pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
           'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pd) ));
WR10: SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT.OF_SHAPE') |
         (sa.description <> 'circular offset') OR
         (SIZEOF( QUERY( pdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
```

```
        ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN
          TYPEOF(r_i)) AND
              ('AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN
           TYPEOF(r_i\measure_representation_item.value_component)) AND
        (r_i.name = 'offset index') )) = 1 )
       )) = 1)
              )) = SIZEOF(USEDIN( pds, 'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE')))
            )) = SIZEOF( QUERY(pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') |
          'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pd) ));
WR11: SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
         TYPEOF(pds)) AND
       (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
             'SHAPE_ASPECT.OF_SHAPE') |
        (sa.description <> 'circular offset') OR
        (SIZEOF( QUERY( pdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
         (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
              (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
             'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
              * TYPEOF(r_i)) = 2) AND
              (r_i.name = 'angular offset') )) = 1 )
        )) = 1 )
             )) = SIZEOF(USEDIN( pds, 'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE')))
            )) = SIZEOF( QUERY(pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') |
          'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pd) ));
WR12: SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
         TYPEOF(pds)) AND
       (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
             'SHAPE_ASPECT.OF_SHAPE') |
        (sa.description <> 'rectangular offsert') OR
        (SIZEOF( QUERY( pdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
         (SIZEOF( pdr.items) = 3)
       )) = 1)
              )) = SIZEOF(USEDIN( pds, 'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE')))
            )) = SIZEOF( QUERY(pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') |
          'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pd) ));
WR13: SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
```

```
            'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT.OF_SHAPE') |
         NOT (sa.description IN ['rectangular omit',
         'rectangular offset']) OR
         (SIZEOF( QUERY( pdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
          ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN
            TYPEOF(r_i)) AND
          ('AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN
        TYPEOF(r_i\measure_representation_item.value_component)) AND
          (r_i.name = 'row index') )) = 1 )
        )) = 1)
              )) = SIZEOF(USEDIN( pds, 'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE')))
            )) = SIZEOF( QUERY(pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
           'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pd) ));
WR14: SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT.OF_SHAPE') |
         NOT (sa.description IN ['rectangular omit',
         'rectangular offset']) OR
         (SIZEOF( QUERY( pdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
          ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN
            TYPEOF(r_i)) AND
          ('AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN
            TYPEOF(r_i\measure_representation_item.value_component)) AND
          (r_i.name = 'column index') )) = 1 )
        )) = 1)
              )) = SIZEOF(USEDIN( pds, 'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE')))
            )) = SIZEOF( QUERY(pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
           'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
            TYPEOF(pd) ));
WR15: SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
```

```
              'SHAPE_ASPECT.OF_SHAPE') |
          (sa.description <> 'rectangular offset') OR
          (SIZEOF( QUERY( pdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
             TYPEOF(pdr.used_representation)) AND
           (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
           (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
            'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
           * TYPEOF(r_i)) = 2) AND
           (r_i.name = 'offset distance') )) = 1 )
         )) = 1)
               )) = SIZEOF(USEDIN( pds, 'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE')))
              )) = SIZEOF( QUERY(pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION.DEFINITION') |
            'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd) ));
  WR16: SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pds)) AND
         (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT.OF_SHAPE') |
          (sa.description <> 'rectangular offset') OR
          (SIZEOF( QUERY( pdr <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN
             TYPEOF(pdr.used_representation)) AND
           (pdr.used_representation.description = 'offset direction')
         )) = 1)
               )) = SIZEOF(USEDIN( pds, 'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE')))
              )) = SIZEOF( QUERY(pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION.DEFINITION') |
            'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd) ));
END_ENTITY;
(*
```

Formal propositions:

WR1: The **modified_pattern** shall be the **related_object** in exactly one **characterized_object_relationship** with a **name** of 'replicate base assignment' in which the **relating_object** is either a **circular_pattern** or a **rectangular_pattern.**

WR2: If the **modified_pattern** is the **related_object** in a **characterized_object_relationship** with a **name** of 'replicate base assignment' in which the **relating_object** is a **circular_pattern,** its shape shall only have **shape_aspect** instances with a **description** of either 'circular omit' or 'circular offset'.

WR3: If the **modified_pattern** is the **related_object** in a **characterized_object_relationship** with a **name** of 'replicate base assignment' in which the **relating_object** is a **circular_-**

pattern, its shape shall have at least one **shape_aspect** with a **description** of either 'circular omit' or 'circular offset'.

WR4: If the **modified_pattern** is the **related_object** in a **characterized_object_relationship** with a **name** of 'replicate base assignment' in which the **relating_object** is a **rectangular_pattern,** its shape shall only have **shape_aspect** instances with a **description** of either 'rectangular omit' or 'rectangular offset'.

WR5: If the **modified_pattern** is the **related_object** in a **characterized_object_relationship** with a **name** of 'replicate base assignment' in which the **relating_object** is a **rectangular_pattern,** its shape shall have at least one **shape_aspect** with a **description** of either 'rectangular omit' or 'rectangular offset'.

WR6: Each **shape_aspect** of the shape of a **modified_pattern** shall have exactly one implicit representation.

WR7: If the **shape_aspect** of the shape of a **modified_pattern** has a **description** of 'circular omit', its implicit representation shall contain exactly one **representation_item** in its set of items.

WR8: If the **shape_aspect** of the shape of a **modified_pattern** has a **description** of 'circular omit', its implicit representation shall contain exactly one **representation_item** of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'omit index'.

WR9: If the **shape_aspect** of the shape of a **modified_pattern** has a **description** of 'circular offset' or 'rectangular omit', its implicit representation shall contain exactly two **representation_items** in its set of items.

WR10: If the **shape_aspect** of the shape of a **modified_pattern** has a **description** of 'circular offset', its implicit representation shall contain exactly one **representation_item** of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'offset index'.

WR11: If the **shape_aspect** of the shape of a **modified_pattern** has a **description** of 'circular offset', its implicit representation shall contain exactly one **representation_item** of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'angular offset'.

WR12: If the **shape_aspect** of the shape of a **modified_pattern** has a **description** of 'rectangular offset', its implicit representation shall contain exactly three **representation_items** in its set of items.

WR13: If the **shape_aspect** of the shape of a **modified_pattern** has a **description** of 'rectangular omit' or 'rectangular offset', its implicit representation shall contain exactly one **representation_item** of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'row index'.

WR14: If the **shape_aspect** of the shape of a **modified_pattern** has a **description** of 'rectangular omit' or 'rectangular offset', its implicit representation shall contain exactly one **representation_item** of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'column index'.

WR15: If the **shape_aspect** of the shape of a **modified_pattern** has a **description** of 'rectangular offset', its implicit representation shall contain exactly one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'offset distance'.

WR16: If the **shape_aspect** of the shape of a **modified_pattern** has a **description** of 'rectangular offset', it shall have exactly one **direction_shape_representation** with a **description** of 'offset direction'.

5.2.3.1.90 ngon_closed_profile

An **ngon_closed_profile** is a type of **shape_aspect** that consists of an enclosed 2D area and a reference system for the parameters of the **ngon_closed_profile.** The enclosed area is defined by three or more straight sides with the corners lying on one circle. The reference system is used to define location and orientation of the **square_u_profile** and consists of two orthogonal reference axes X and Y that intersect in a reference point. See ARM definition for Ngon_profile in paragraph 4.2.288 for more information.

EXPRESS specification:

```
*)
ENTITY ngon_closed_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)
         )) = 1;
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
           ({3 <= SIZEOF(pdr.used_representation.items) <= 4})
              )) = 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         NOT (srwp_i.name IN ['orientation','number of sides',
                                              'circumscribed diameter','corner radius'])
                      )) > 0 )
             )) = 0;
  WR5:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
```

```
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i)) AND
                         (srwp_i.name = 'orientation')
                       )) = 1 )
             )) = 1;
  WR6:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i))
        AND
                ('AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN
                 TYPEOF(srwp_i\measure_representation_item.value_component)) AND
                 (srwp_i.name = 'number of sides')
                       )) = 1 )
             )) = 1;
  WR7:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
           (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
              (srwp_i.name = 'circumscibed diameter') AND
              (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(srwp_i)
          ) = 2 )
            )) = 1 )
        )) = 1;
  WR8:  (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
           (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
              (srwp_i.name = 'corner radius')
           )) > 0 )
              )) = 0 ) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
           (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
              (srwp_i.name = 'corner radius') AND
              (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                 'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(srwp_i)
          ) = 2 )
            )) = 1 )
             )) = 1 );
END_ENTITY;
(*
```

Formal propositions:

WR1: The **ngon_closed_profile** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **ngon_closed_profile** shall have exactly one implicit representation defined by a **shape_representation_with_parameters.**

WR3: The **shape_representation_with_parameters** that represents the **ngon_closed_profile** shall contain three or four **representation_items** in its set of **items**.

WR4: The implicit representation of an **ngon_closed_profile** shall contain only **representation_items** in its set of **items** that have a **name** of either 'orientation', 'number of sides', 'circumscribed diameter', or 'corner radius'.

WR5: Exactly one **representation_item** used for the implicit representation of an **ngon_closed_profile** shall be of type **placement** with a **name** of 'orientation'.

WR6: Exactly one **representation_item** used for the implicit representation of an **ngon_closed_profile** shall be of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'number of sides'.

WR7: Exactly one **representation_item** used for the implicit representation of an **ngon_closed_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'circumscribed diameter'.

WR8: At most one **representation_item** used for the implicit representation of an **ngon_closed_profile** shall have a **name** of 'corner radius'. If present, this **representation_item** shall be of type **measure_representation_item** and **length_measure_with_unit.**

Informal propositions:

IP1: The **ngon_closed_profile** shall be planar and shall lie in the X-Y plane of the reference system.

IP2: The reference point of the **ngon_closed_profile** is defined at the centre of the circumscribed circle.

IP3: One of the sides of the ngon shall be parallel to the X direction intersecting the negative Y direction.

5.2.3.1.91 occurrence_usage_definition

An **occurrence_usage_definition** is a type of **configuration_design** that identifies a **configurable_item** that serves as definition for a particular **product_definition_occurrence.** See ARM definition for Item_instance in paragraph 4.2.241 for more information.

EXPRESS specification:

```
*)
```

```
ENTITY occurrence_usage_definition
  SUBTYPE OF (configuration_design);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_OCCURRENCE' IN TYPEOF(SELF.design);
  WR2:  ('AUTOMOTIVE_DESIGN.CONFIGURABLE_ITEM' IN TYPEOF(SELF.configuration)) AND
        (SELF.configuration.description = 'product specification');
END_ENTITY;
(*
```

Formal propositions:

WR1: The **occurrence_usage_definition** shall reference as the **design** a **product_definition** that is of type **product_definition_occurrence.**

WR2: The **occurrence_usage_definition** shall reference as the **configuration** a **configuration_item** that is of type **configurable_item** and has a **description** of 'product specification'.

5.2.3.1.92 open_path_profile

An **open_path_profile** is a type of **shape_aspect** that consists of a connected set of curves and possibly a reference system that may be used to place the **open_path_profile.** The start vertex and end vertex for the set of curves do not connect creating an area open on one side. The reference system consists of two orthogonal reference axes X and Y that intersect in a reference point. See ARM definition for General_open_profile in paragraph 4.2.202 for more information.

EXPRESS specification:

```
*)
ENTITY open_path_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)
         )) <= 1;
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
           (SIZEOF(pdr.used_representation.items) = 1)
              )) <= 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
```

```
                        (srwp_i.name = 'orientation') AND
                        ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i))
                     )) = 1)
             )) = 1;
  WR5:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.PATH_SHAPE_REPRESENTATION' IN
           TYPEOF(pdr.used_representation))
             )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **open_path_profile** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **open_path_profile** shall have at most one implicit representation defined by a **shape_representation_with_parameters.**

WR3: The **shape_representation_with_parameters** that represents the **open_path_profile** shall contain one **representation_item** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of an **open_path_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: The **open_path_profile** shall have exactly one **path_shape_representation.**

Informal propositions:

IP1: The **open_path_profile** shall be planar and shall lie in the X-Y plane of the reference system if this is present.

5.2.3.1.93 package_product_concept_feature

A **package_product_concept_feature** is a type of **product_concept_feature** that combines several **product_concept_feature** instances that shall be offered to the market as a set. See ARM definition for Specification in paragraph 4.2.415 for more information.

EXPRESS specification:

```
*)
ENTITY package_product_concept_feature
  SUBTYPE OF (product_concept_feature);
END_ENTITY;
(*
```

5.2.3.1.94 partial_circular_profile

A **partial_circular_profile** is a type of **shape_aspect** that consists of an arc of constant radius and a reference system for the parameters of the **partial_circular_profile.** The reference system is used to define location and orientation of the **partial_circular_profile** and consists of two orthogonal reference axes X and Y that intersect in a reference point. See ARM definition for Partial_circular_profile in paragraph 4.2.303 for more information.

EXPRESS specification:

```
*)
ENTITY partial_circular_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)
         )) = 1;
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
           (SIZEOF(pdr.used_representation.items) = 3)
              )) <= 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'orientation') AND
                         ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i))
                      )) = 1)
             )) = 1;
  WR5:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'radius') AND
                         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                  'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                  * TYPEOF(srwp_i)) = 2 )
                      )) = 1)
             )) = 1;
  WR6:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
```

```
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'sweep angle') AND
                         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                  'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
                                  * TYPEOF(srwp_i)) = 2 )
                       )) = 1)
            )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **partial_circular_profile** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **partial_circular_profile** shall have exactly one implicit representation defined by a **shape_representation_with_parameters.**

WR3: The **shape_representation_with_parameters** that represents the **partial_circular_profile** shall contain three **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **partial_circular_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: Exactly one **representation_item** used for the implicit representation of a **partial_circular_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR6: Exactly one **representation_item** used for the implicit representation of a **partial_circular_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'sweep angle'.

Informal propositions:

IP1: The **partial_circular_profile** shall be planar and shall lie in the X-Y plane of the reference system.

IP2: The reference point of the **partial_circular_profile** is defined at the arc origin.

IP3: The arc that defines the **partial_circular_profile** shall start at the X axis.

5.2.3.1.95 path_feature_component

A **path_feature_component** is a type of **shape_aspect** that represents a path used to define preconceived form patterns. The **path_feature_component** may be represented explicitly by a curve or a set of curves or implicitly by parameters that are defined in a reference system. The reference system is an **axis_placement.** See ARM definition for Path in paragraph 4.2.304, for General_path in paragraph 4.2.203, for Linear_path in paragraph 4.2.268, for Circular_path in paragraph 4.2.60, for Complete_circular_path in paragraph 4.2.78, and for Partial_circular_path in paragraph 4.2.302 for more information.

EXPRESS specification:

```
*)
ENTITY path_feature_component
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SELF.description IN ['partial circular','complete circular',
                             'linar','complex'];
  WR3:  (SELF.description <> 'complex') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                   'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation))
               )) = 0);
  WR4:  NOT (SELF.description IN ['partial circular','complete circular','linar']) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                   'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
                 (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                          (srwp_i.name = 'orientation') AND
                          ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i))
                        )) = 1)
               )) = 1);
  WR5:  (SELF.description <> 'partial circular') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                   'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
                 (SIZEOF(pdr.used_representation.items) = 3)
               )) = 1);
  WR6:  (SELF.description <> 'partial circular') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                   'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
                 (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                          (srwp_i.name = 'radius') AND
                          (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                   'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                  * TYPEOF(srwp_i)) = 2 )
                        )) = 1)
               )) = 1);
  WR7:  (SELF.description <> 'partial circular') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                   'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
                 (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                          (srwp_i.name = 'sweep angle') AND
```

```
                        (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                 'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
                                 * TYPEOF(srwp_i)) = 2 )
                     )) = 1)
             )) = 1);
WR8:  (SELF.description <> 'complete circular') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
               (SIZEOF(pdr.used_representation.items) = 2)
             )) = 1);
WR9:  (SELF.description <> 'complete circular') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
               (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                        (srwp_i.name = 'radius') AND
                        (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                 'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                 * TYPEOF(srwp_i)) = 2 )
                     )) = 1)
             )) = 1);
WR10: (SELF.description <> 'linear') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
               (SIZEOF(pdr.used_representation.items) = 2)
             )) = 1);
WR11: (SELF.description <> 'linear') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
               (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                        (srwp_i.name = 'distance') AND
                        (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                 'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                 * TYPEOF(srwp_i)) = 2 )
                     )) = 1)
             )) = 1);
WR12: (SELF.description <> 'complex') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PATH_SHAPE_REPRESENTATION' IN
          TYPEOF(pdr.used_representation)) AND
               (pdr.used_representation.description = 'path geometry') AND
               (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                        (srwp_i.name = 'path curve') AND
                        ('AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF(srwp_i))
```

```
                        )) = 1)
                )) = 1);
END_ENTITY;
(*
```

Formal propositions:

WR1: The **path_feature_component** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **description** for the **path_feature_component** shall be either 'partial circular', 'complete circular', 'linear', or 'complex'.

WR3: If the **description** of the **path_feature_component** is 'complex' the **path_feature_component** shall not have an implicit representation specified by a **shape_representation_with_parameters.**

WR4: If the **description** of the **path_feature_component** is 'linear', 'partial circular', or 'complete circular' the **path_feature_component** shall have an implicit representation specified by a **shape_representation_with_parameters.** This **shape_representation_with_parameters** shall contain exactly one **representation_item** of type **placement** with a **name** of 'orientation'.

WR5: If the **description** of the **path_feature_component** is 'partial circular', the implicit representation of the **path_feature_component** shall contain exactly three **representation_items** in its set of **items**.

WR6: If the **description** of the **path_feature_component** is 'partial circular', exactly one **representation_item** used for the implicit representation of the **path_feature_component** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR7: If the **description** of the **path_feature_component** is 'partial circular', exactly one **representation_item** used for the implicit representation of the **path_feature_component** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'sweep angle'.

WR8: If the **description** of the **path_feature_component** is 'complete circular', the implicit representation of the **path_feature_component** shall contain exactly two **representation_items** in its set of **items**.

WR9: If the **description** of the **path_feature_component** is 'complete circular', exactly one **representation_item** used for the implicit representation of the **path_feature_component** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR10: If the **description** of the **path_feature_component** is 'linear', the implicit representation of the **path_feature_component** shall contain exactly two **representation_items** in its set of **items**.

WR11: If the **description** of the **path_feature_component** is 'linear', exactly one **representation_item** used for the implicit representation of the **path_feature_component** shall

be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'distance'.

WR12: If the **description** of the **path_feature_component** is 'complex', the **path_feature_component** shall have exactly one **path_shape_representation** with a **description** of 'path geometry'. This **path_shape_representation** shall contain a **curve** with a **name** of 'path curve'.

Informal propositions:

IP1: In the case of a linear **path_feature_component** the path starts at the intersection of the three directions of the **axis2_placement_3d** and travels along the third direction.

IP2: In the case of a circular **path_feature_component** the path lies in the plane defined by the first and second direction of the **axis2_placement_3d;** the third direction of the **axis2_placement_3d** is the rotation axis.

5.2.3.1.96 path_shape_representation

A **path_shape_representation** is a type of **representation** that represents a curved path such that the collection of curves is continuous and directed.

EXPRESS specification:

```
*)
ENTITY path_shape_representation
  SUBTYPE OF (representation);
WHERE
  WR1:  SIZEOF(SELF.items) = 1;
  WR2:  SIZEOF( QUERY( i <* SELF.items |
                'AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF(i) )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **path_shape_representation** shall have exactly one **representation_item** in its set of **items**.

WR2: The geometric element that is used to represent the **path_shape_representation** shall be a **curve**.

5.2.3.1.97 physically_modelled_product_definition

A **physically_modelled_product_definition** is a type of **product_definition_with_associated_documents** that specifies a model of a shape of the part which is physically available. See ARM definition for Physical_model in paragraph 4.2.312 for more information.

EXPRESS specification:

```
*)
ENTITY physically_modelled_product_definition
  SUBTYPE OF (product_definition_with_associated_documents);
WHERE
  WR1:  SELF.frame_of_reference\application_context_element.name = 'physical occurrence';
  WR2:  (SIZEOF(SELF.documentation_ids) = 1) AND
        (SIZEOF( QUERY( d <* SELF.documentation_ids |
                 d.kind.product_data_type = 'physical model' )) = 1);
  WR3:  SIZEOF( QUERY( aoa <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
                aoa.role.name = 'location'
              )) = 1;
  WR4:  (SIZEOF( QUERY( pd <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
                                        'PROPERTY_DEFINITION.DEFINITION') |
                           'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
                            TYPEOF(pd))) = 0 ) OR
        (SIZEOF( QUERY ( pds <* USEDIN( SELF,
                         'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.DEFINITION') |
                 ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
                   TYPEOF(pds)) AND
                 (SIZEOF( USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
                          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1 ) AND
                 (SIZEOF( QUERY( pdr <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
                                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                          ('AUTOMOTIVE_DESIGN.SHAPE_DEFINITION_REPRESENTATION' IN
                           TYPEOF(pdr)) AND
                          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN
                           TYPEOF(pdr.used_representation)) AND
                          (SIZEOF(pdr.used_representation.items) = 1)
                        )) = 1 )
               )) = 1 );
END_ENTITY;
(*
```

Formal propositions:

WR1: The **physically_modelled_product_definition** shall reference as its **frame_of_reference** a **product_definition_context** that has a **name** of 'physical occurrence'.

WR2: The **physically_modelled_product_definition** shall have exactly one **document** in its set of **documentation_ids.** This **document** shall reference as its **kind** a **document_type** that has a **product_data_type** of 'physical model'.

WR3: The **physically_modelled_product_definition** shall be contained in the set of items of exactly one **applied_organization_assignment** that references as its **role** an **organization_role** that has a **name** of 'location'.

WR4: If the **physically_modelled_product_definition** is referenced as **definition** by a **product_definition_shape,** this **product_definition_shape** shall be referenced as the **defini-**

tion by exactly one instance of **property_definition_representation.** This **property_definition_representation** shall be a **shape_definition_representation** and shall reference as its **representation** a **shape_representation** that contains exactly one **representation_item** in its set of **items**. This **representation_item** shall be a **measure_representation_item** and a **ratio_measure_with_unit** and shall have a **name** of 'scale value'.

Informal propositions:

IP1: In case the **physically_modelled_product_definition** is not referenced by a **product_definition_shape** as the **definition**, the scale shall be 1:1.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **physically_modelled_product_definition** entity:

– restrict_version_assignment_for_physically_modelled_product_definition (see 5.2.4.173).

5.2.3.1.98 placed_datum_target_feature

A **placed_datum_target_feature** is a type of **datum_target** that represents the implicit definition. See ARM definition for Placed_target in paragraph 4.2.314 for more information.

EXPRESS specification:

```
*)
ENTITY placed_datum_target_feature
  SUBTYPE OF (datum_target);
WHERE
  WR1:  SELF.description IN ['point', 'line', 'rectangle', 'circle'];
  WR2:  SIZEOF (QUERY (pd <* USEDIN (SELF,
        'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
        'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation))) = 1))) = 0;
  WR3:   SIZEOF (QUERY (pd <* USEDIN (SELF,
        'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.' +
        'DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF (it)) AND
```

```
       (it.name = 'orientation'))) = 1))) = 0))) = 0;
 WR4:  (NOT (SELF.description = 'point')) OR
       (SIZEOF (QUERY (pd <* USEDIN (SELF,
       'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.' +
       'DEFINITION') |
       NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
       'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       'AUTOMOTIVE_DESIGN.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS'
       IN TYPEOF (pdr.used_representation)) |
       NOT (SIZEOF (impl_rep.used_representation.items) = 1))) = 0)))  = 0);
 WR5:  (NOT (SELF.description IN ['line', 'circle'])) OR
       (SIZEOF (QUERY (pd <* USEDIN (SELF,
       'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.' +
       'DEFINITION') |
       NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
       'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       'AUTOMOTIVE_DESIGN.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS'
       IN TYPEOF (pdr.used_representation)) |
       NOT (SIZEOF (impl_rep.used_representation.items) = 2))) = 0)))  = 0);
 WR6:  (NOT (SELF.description = 'rectangle')) OR
       (SIZEOF (QUERY (pd <* USEDIN (SELF,
       'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.' +
       'DEFINITION') |
       NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
       'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       'AUTOMOTIVE_DESIGN.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS'
       IN TYPEOF (pdr.used_representation)) |
       NOT (SIZEOF (impl_rep.used_representation.items) = 3))) = 0))) = 0);
 WR7:  (NOT (SELF.description = 'circle')) OR
       (SIZEOF (QUERY (pd <* USEDIN (SELF,
       'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
       'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       'AUTOMOTIVE_DESIGN.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS'
       IN TYPEOF (pdr.used_representation)) |
       NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
       (SIZEOF (['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF (it)) = 2) AND
       (it.name = 'target diameter'))) = 1))) = 0))) = 0);
 WR8:  (NOT (SELF.description = 'line')) OR
       (SIZEOF (QUERY (pd <* USEDIN (SELF,
       'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.DEFINITION') |
       NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
```

```
        'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        (SIZEOF (['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND (it.name = 'target length'))) = 1))) = 0))) = 0);
  WR9:  (NOT (SELF.description = 'rectangle')) OR
        (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        (SIZEOF (['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND (it.name = 'target length'))) = 1))) = 0))) = 0);
  WR10: (NOT (SELF.description = 'rectangle')) OR
        (SIZEOF (QUERY (pd <* USEDIN (SELF,
        'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.DEFINITION') |
        NOT (SIZEOF (QUERY (impl_rep <* QUERY (pdr <* USEDIN (pd,
        'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF (pdr.used_representation)) |
        NOT (SIZEOF (QUERY (it <* impl_rep.used_representation.items |
        (SIZEOF (['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF (it)) = 2) AND (it.name = 'target width'))) = 1))) = 0))) = 0);
END_ENTITY;
(*
```

Formal propositions:

WR1: The description for the **placed_datum_target_feature** shall be either 'point', 'line', 'rectangle' or 'circle'.

WR2: A **placed_datum_target_feature** shall have exactly one implicit representation.

WR3: Exactly one **representation_item** used for the implicit representation of the **placed_datum_target_feature** shall be of type placement with a name of 'orientation'.

WR4: If the **placed_datum_target_feature** is a point, the representation shall contain exactly one **representation_items** in its set of items.

WR5: If the **placed_datum_target_feature** is a line or circle, the representation shall contain exactly two **representation_items** in its set of items.

WR6: If the **placed_datum_target_feature** is a rectangle, the representation shall contain exactly three **representation_items** in its set of items.

WR7: If the description of the **placed_datum_target_feature** is 'circle', exactly one **representation_item** used for the implicit representation of the **placed_datum_target_feature** shall be of type **measure_representation_item** and **length_measure_with_unit** with a name of 'target diameter'.

WR8: If the description of the **placed_datum_target_feature** is 'line', exactly one **representation_item** used for the implicit representation of the **placed_datum_target_feature** shall be of type **measure_representation_item** and **length_measure_with_unit** with a name of 'target length'.

WR9: If the description of the **placed_datum_target_feature** is 'rectangle', exactly one **representation_item** used for the implicit representation of the **placed_datum_target_feature** shall be of type **measure_representation_item** and **length_measure_with_unit** with a name of 'target length'.

WR10: If the description of the **placed_datum_target_feature** is 'rectangle', exactly one **representation_item** used for the implicit representation of the **placed_datum_target_feature** shall be of type **measure_representation_item** and **length_measure_with_unit** with a name of 'target width'.

5.2.3.1.99 placed_feature

A **placed_feature** is a type of **shape_aspect** that is the identification of a preconceived form pattern on a part as a result of the placement of a **feature_definition** which is defined in its own context. See ARM definition for Placed_feature in paragraph 4.2.313 for more information.

EXPRESS specification:

```
*)
ENTITY placed_feature
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  SIZEOF(['AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION',
                'AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_SOLID',
                'AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_PANEL']
               * TYPEOF(SELF.of_shape.definition)) = 1;
  WR2:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                (pdr.used_representation.name = 'feature definition placement') AND
         (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                         ('AUTOMOTIVE_DESIGN.MAPPED_ITEM' in TYPEOF(r_i)) AND
                         (SIZEOF( QUERY( pdr <* USEDIN(r_i\mapped_item.
                                                 mapping_source.mapped_representation,
                                             'AUTOMOTIVE_DESIGN.' +
```

```
                                        'PROPERTY_DEFINITION_REPRESENTATION.' +
                                        'USED_REPRESENTATION') |
                                  'AUTOMOTIVE_DESIGN.FEATURE_DEFINITION' IN
                                   TYPEOF(pdr.definition\property_definition.
                                          definition)
                              )) = 1 )
                       )) = 1 )
               )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **placed_feature** shall be an aspect of the shape of either a **product_definition,** a **compound_feature_in_solid,** or a **compound_feature_in_panel.**

WR2: The **placed_feature** shall have a **shape_representation** with a **description** of 'feature definition placement' that contains a **mapped_item** in its set of **items**. This **mapped_item** shall reference a **representation_map** that references the **shape_representation** of a **feature_definition** as **mapped_representation.**

5.2.3.1.100 pocket

A **pocket** is a type of **feature_definition** that is the representation of a volume that is removed from a solid base shape. This volume may be enclosed in which case it is represented by sweeping a closed profile along a linear path represented by a **path_feature_component.** If the removed volume is not enclosed it is represented by sweeping an open profile. The bottom of the pocket is represented by a **pocket_bottom.** The walls of a **pocket** may have a slope that is represented by a **taper**. See ARM definition for General_pocket_in_solid in paragraph 4.2.205, Rectangular_closed_pocket_in_solid in paragraph 4.2.369, and Rectangular_open_pocket_in_solid in paragraph 4.2.373 for more information.

EXPRESS specification:

```
*)
ENTITY pocket
  SUBTYPE OF (feature_definition);
WHERE
  WR1:  SELF.description IN ['open rectangular','closed rectangular',
              'non-rectangular'];
  WR2:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pds)) AND
         (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
                'SHAPE_ASPECT.OF_SHAPE') |
          (sa.name = 'sweep path') AND
          (sa.description = 'path feature component occurrence') AND
          (SIZEOF( QUERY( sdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
```

```
          'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
           ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
             TYPEOF(sdr)) AND
           (sdr.name = 'path feature component usage') AND
           ('AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN
             TYPEOF(sdr.relating_shape_aspect)) AND
           (sdr.relating_shape_aspect.description = 'linear')
          )) = 1 )
               )) = 1 )
        )) = 1;
WR3:  NOT (SELF.description = 'non-rectangular') OR
      (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pds)) AND
         (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT.OF_SHAPE') |
           (sa.name = 'boundary') AND
           (sa.description = 'profile occurrence') AND
           (SIZEOF( QUERY( sdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
             'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
             ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
               TYPEOF(sdr)) AND
             (sdr.name = 'profile usage') AND
             (SIZEOF(['AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE',
              'AUTOMOTIVE_DESIGN.NGON_CLOSED_PROFILE',
              'AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE',
              'AUTOMOTIVE_DESIGN.CLOSED_PATH_PROFILE',
              'AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE',
              'AUTOMOTIVE_DESIGN.PARTIAL_CIRCULAR_PROFILE',
              'AUTOMOTIVE_DESIGN.ROUNDED_U_PROFILE',
              'AUTOMOTIVE_DESIGN.VEE_PROFILE',
              'AUTOMOTIVE_DESIGN.TEE_PROFILE',
              'AUTOMOTIVE_DESIGN.OPEN_PATH_PROFILE'] *
              TYPEOF(sdr.relating_shape_aspect)
              ) = 1 )
          )) = 1 )
        )) = 1 )
              )) = 1 );
WR4:  NOT (SELF.description = 'closed rectangular') OR
      (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
           TYPEOF(pds)) AND
         (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT.OF_SHAPE') |
         (sa.name = 'boundary') AND
         (sa.description = 'closed profile occurrence') AND
         (SIZEOF( QUERY( sdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
            TYPEOF(sdr)) AND
```

```
          (sdr.name = 'closed profile usage') AND
          ('AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE' IN
           TYPEOF(sdr.relating_shape_aspect))
         )) = 1 )
        )) = 1 )
             )) = 1 );
WR5:  NOT (SELF.description = 'open rectangular') OR
      (SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
             'SHAPE_ASPECT.OF_SHAPE') |
        (sa.name = 'boundary') AND
        (sa.description = 'closed profile occurrence') AND
        (SIZEOF( QUERY( sdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
          TYPEOF(sdr)) AND
        (sdr.name = 'closed profile usage') AND
        ('AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE' IN
         TYPEOF(sdr.relating_shape_aspect))
       )) = 1 )
       )) = 1 )
             )) = 1 );
WR6:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT.OF_SHAPE') |
        (sa.name = 'bottom condition') AND
        (sa.description = 'pocket bottom occurrence') AND
        (SIZEOF( QUERY( fcr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
         ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
           TYPEOF(fcr)) AND
         (fcr.name = 'pocket bottom usage') AND
         ('AUTOMOTIVE_DESIGN.POCKET_BOTTOM' IN
          TYPEOF(fcr.relating_shape_aspect))
        )) = 1 )
        )) = 1 )
       )) = 1;
WR7:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT.OF_SHAPE') |
        (sa.name = 'taper') AND
        (sa.description = 'taper occurrence') AND
        (SIZEOF( QUERY( fcr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
```

```
          'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(fcr)) AND
          (fcr.name = 'taper usage') AND
          ('AUTOMOTIVE_DESIGN.TAPER' IN
           TYPEOF(fcr.relating_shape_aspect))
         )) = 1 )
        )) = 1 )
        )) <= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **pocket** shall have a description of 'open rectangular', 'closed rectangular', or 'non-rectangular'.

WR2: If the **pocket** has a description of 'non-rectangular', the **pocket** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'boundary' and a **description** of 'profile occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'profile usage'in which the **relating_shape_aspect** is a **circular_closed_profile, ngon_closed_profile, rectangular_closed_profile, closed_path_profile, square_u_profile, partial_circular_profile, rounded_u_profile, vee_profile, tee_profile,** or **open_path_profile.**

WR3: If the **pocket** has a description of 'closed rectangular', the **pocket** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'boundary' and a **description** of 'closed profile occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'closed profile usage' in which the **relating_shape_aspect** is a **rectangular_closed_profile.**

WR4: If the **pocket** has a description of 'open rectangular', the **pocket** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'boundary' and a **description** of 'open profile occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'open profile usage' in which the **relating_shape_aspect** is a **square_u_profile.**

WR5: The **pocket** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'sweep path' and a **description** of 'path feature component occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'path feature component usage' in which the **relating_shape_aspect** is a **path_feature_component** with a **description** of 'linear'.

WR6: The **pocket** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'bottom condition' and a **description** of 'pocket bottom occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'pocket bottom usage' in which the **relating_shape_aspect** is a **pocket_bottom.**

WR7: The **pocket** shall be the basis shape for at most one **shape_aspect** with a **name** of 'taper' and a **description** of 'taper occurrence'. If present, this **shape_aspect** shall be the **re-**

lated_shape_aspect in exactly one **feature_component_relationship** with a **name** of 'taper usage' in which the **relating_shape_aspect** is a **taper**.

5.2.3.1.101 pocket_bottom

A **pocket_bottom** is a type of **shape_aspect** that is the bottom condition for a **pocket**. See ARM definition for Pocket_bottom_condition in paragraph 4.2.320, for Planar_bottom_condition in paragraph 4.2.315, for Through_bottom_condition in paragraph 4.2.459, and for General_-bottom_condition in paragraph 4.2.191 for more information.

EXPRESS specification:

```
*)
ENTITY pocket_bottom
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
        TYPEOF(SELF.of_shape.definition);
  WR2:  SELF.description IN ['planar','non-planar','through'];
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)
               )) = 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
          (SIZEOF(pdr.used_representation.items) = 1) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
           ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
             TYPEOF(srwp_i)) AND
            (srwp_i.name = 'bottom location') AND
            (srwp_i.description IN ['path start','path end'])
          )) = 1 )
               )) = 1;
  WR5:  (SELF.description <> 'non-planar') OR
        ((SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            (SIZEOF(['AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION',
             'AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION'] *
              TYPEOF(pdr.used_representation)) = 1 )
          )) = 1 ) AND
         (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            (SIZEOF(['AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION',
             'AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION']
             * TYPEOF(pdr.used_representation)) = 1 ) AND
            (pdr.used_representation.description = 'floor face')
         )) = 1 ));
```

```
WR6:  (SELF.description <> 'planar') OR
      ((SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.LOCATION_SHAPE_REPRESENTATION' IN
           TYPEOF(pdr.used_representation))
       )) = 1 ) AND
       (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.LOCATION_SHAPE_REPRESENTATION' IN
           TYPEOF(pdr.used_representation)) AND
          (pdr.used_representation.description = 'floor location')
       )) = 1 ));
WR7:  (SELF.description <> 'planar') OR
      ((SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN
          TYPEOF(pdr.used_representation))
       )) = 1 ) AND
       (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN
          TYPEOF(pdr.used_representation)) AND
         (pdr.used_representation.description = 'floor normal')
       )) = 1 ));
WR8:  (SELF.description <> 'through') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        (SIZEOF(pdr.used_representation.items) = 1)
             )) = 1);
WR9:  NOT (SELF.description IN ['planar','non-planar']) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        ({1 <= SIZEOF(pdr.used_representation.items) <= 2})
             )) = 1 );
WR10: NOT (SELF.description IN ['planar','non-planar']) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
           'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
          * TYPEOF(srwp_i)) = 2) AND
          (srwp_i.name = 'floor radius')
        )) = 1 )
             )) <= 1 );
WR11: (SIZEOF( QUERY( sar <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
```

```
            'AUTOMOTIVE_DESIGN.FAETURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(sar))) = 1 ) AND
        (SIZEOF( QUERY( fcr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.FAETURE_COMPONENT_RELATIONSHIP' IN
           TYPEOF(fcr)) AND
          (fcr.name = 'pocket bottom usage') AND
          ('AUTOMOTIVE_DESIGN.POCKET' IN
           TYPEOF(fcr.related_shape_aspect.of_shape.definition)) AND
          (fcr.related_shape_aspect.of_shape.definition.name =
           'bottom condition') AND
          (fcr.related_shape_aspect.of_shape.definition.description =
           'pocket bottom occurrence')
              )) = 1 );
END_ENTITY;
(*
```

Formal propositions:

WR1: The **pocket_bottom** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **description** of the **pocket_bottom** shall be either 'planar', 'non-planar', or 'through'.

WR3: The **pocket_bottom** shall have its implicit representation specified by exactly one **shape_representation_with_parameters.**

WR4: The implicit representation of the **pocket_bottom** shall contain exactly one **representation_item** in its set of **items** which is a **descriptive_representation_item** with a **name** of 'bottom location' and a **description** of either 'path start' or 'path end'.

WR5: If the **description** of the **pocket_bottom** is 'non-planar', the **pocket_bottom** shall be represented by exactly one of either **face_shape_representation** or **connected_face_shape_representation.** This representation shall have a **description** of 'floor face'.

WR6: If the **description** of the **pocket_bottom** is 'planar', the **pocket_bottom** shall be represented by exactly one **location_shape_representation.** This representation shall have a **description** of 'floor location'.

WR7: If the **description** of the **pocket_bottom** is 'planar', the **pocket_bottom** shall be represented by exactly one **direction_shape_representation.** This representation shall have a **description** of 'floor normal'.

WR8: If the **description** of the **pocket_bottom** is 'through', the **pocket_bottom** shall be represented implicitly by a **shape_representation_with_parameters** that contains exactly one **representation_item** in its set of **items** .

WR9: If the **description** of the **pocket_bottom** is 'planar' or 'non-planar', the **pocket_bottom** shall be represented implicitly by a **shape_representation_with_parameters** that contains between one and two **representation_items** in its set of **items**.

WR10: If the **description** of the **pocket_bottom** is 'planar' or 'non-planar', the implicit representation of the **pocket_bottom** shall contain at most one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'floor radius'. in its set of **items** .

WR11: The **pocket_bottom** shall be the **relating_shape_aspect** in exactly one **feature_component_relationship.** This **feature_component_relationship** shall have a **name** of 'pocket bottom usage' and shall reference as the **related_shape_aspect** an aspect of the shape of a **pocket** that has a **name** of 'bottom condition' and a **description** of 'pocket bottom occurrence'.

5.2.3.1.102 point_placement_shape_representation

A **point_placement_shape_representation** is a type of **shape_representation** that consists of sequences of pairs of points and directions. See ARM definition for Point_direction_model in paragraph 4.2.325 for more information.

EXPRESS specification:

```
*)
ENTITY point_placement_shape_representation
  SUBTYPE OF (shape_representation);
WHERE
  WR1:  SIZEOF( QUERY( i <* SELF.items |
                SIZEOF(['AUTOMOTIVE_DESIGN.COMPOUND_REPRESENTATION_ITEM',
                        'AUTOMOTIVE_DESIGN.MAPPED_ITEM',
                        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D']
                       * TYPEOF(i)) <> 1
              )) = 0;
  WR2:  SIZEOF( QUERY( i <* SELF.items |
                SIZEOF(['AUTOMOTIVE_DESIGN.COMPOUND_REPRESENTATION_ITEM',
                        'AUTOMOTIVE_DESIGN.MAPPED_ITEM']
                       * TYPEOF(i)) = 1
              )) > 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **items** in a **point_placement_shape_representation** shall be of type **compound_representation_item**, **mapped_item**, or **axis2_placement_3d.**

WR2: At least one of the **items** in a **point_placement_shape_representation** shall be either a **compound_representation_item** or a **mapped_item.**

Informal propositions:

IP1: For each **compound_representation_item** in the set of **items** the **item_element** shall be of type **list_representation_item.**

IP2: For each **compound_representation_item** in the set of **items** the **list_representation_-item** shall include only **representation_item**s of type **placement** or **point**.

5.2.3.1.103 pre_defined_point_marker_symbol

A **pre_defined_point_marker_symbol** is a type of **pre_defined_symbol** that presents the position of a point and is identified by name.

NOTE – The definition of this entity is identical to the definition in ISO 10303–202.

EXPRESS specification:

```
*)
ENTITY pre_defined_point_marker_symbol
  SUBTYPE OF (pre_defined_marker);
WHERE
  WR1:  SELF.name IN
        ['asterisk',
        'circle',
        'dot',
        'plus',
        'square',
        'triangle',
        'x'];
END_ENTITY;
(*
```

Formal propositions:

WR1: The name of the **pre_defined_point_marker_symbol** shall be 'asterisk', 'circle', 'dot', 'plus', 'square', 'triangle', or 'x'.

Attribute value definitions:

See ARM definition for Point_appearance in paragraph 4.2.323 for a figure that defines each **pre_defined_point_marker_symbol** which shall be supported by all implementations of this part of ISO 10303. Each symbol definition includes the size of the symbol prior to scaling. The orientation of the individual symbol is exactly as shown in this figure.

The **pre_defined_point_marker_symbols** are:

asterisk: a graphical symbol depicted as three line segments of equal length that intersect at their midpoints forming the origin of the symbol. One of the segments is parallel to the vertical axis of the coordinate system into which the symbol is placed. The other two segments are at angles of 60 and 120 degrees from the first segment, rotated about the origin.

circle: a graphical symbol depicted as a circle. The origin of the symbol is the geometric centre of the circle.

dot: a graphical symbol depicted as a circle with a fill-pattern applied to it. The origin of the dot symbol is the centre of the circle.

plus: a graphical symbol depicted as two perpendicular line segments. The origin of the symbol is the intersection point of the two lines.

square: a graphical symbol depicted as an even-sided rectangle. The origin of the symbol is the geometrical centre of the rectangle.

triangle: a graphical symbol depicted as three line segments that form an equilateral triangle. The origin of the triangle corresponds to the geometric centre of the triangle. One side of the triangle is parallel to the horizontal axis of the coordinate system into which the symbol is placed.

x: a graphical symbol depicted as two line segments of equal length that intersect at their midpoints forming the origin of the symbol. One line segment is at an angle of 45 degrees to the vertical axis of the coordinate system into which the symbol is placed. The other segment is perpendicular to the first.

5.2.3.1.104 pre_defined_terminator_symbol

A **pre_defined_terminator_symbol** is a type of **pre_defined_symbol** that presents a terminator and is identified by name.

NOTE – The definition of this entity is identical to the definition in ISO 10303–202.

EXPRESS specification:

```
*)
ENTITY pre_defined_terminator_symbol
  SUBTYPE OF (pre_defined_symbol);
WHERE
  WR1:  SELF.name IN ['blanked arrow', 'blanked box',
                      'blanked dot', 'dimension origin', 'filled arrow',
                      'filled box', 'filled dot', 'integral symbol',
                      'open arrow', 'slash', 'unfilled arrow'];
END_ENTITY;
(*
```

Formal propositions:

WR1: The name of the **pre_defined_terminator_symbol** shall be 'blanked arrow', 'blanked box', 'blanked dot', 'dimension origin', 'filled arrow', 'filled box', 'filled dot', 'integral symbol', 'open arrow', 'slash', or 'unfilled arrow'.

Attribute value definitions:

See ARM definition for Annotation_symbol in paragraph 4.2.18 for a figure that defines each **pre_defined_terminator_symbol** which shall be supported by all implementations of this part of ISO 10303. Each symbol definition includes the size of the symbol prior to scaling. The orientation of each symbol as positioned on an annotation curve parallel to the x-axis is zero degrees. The annotation curves shown in the figure are not part of the symbol definition.

The **pre_defined_terminator_symbols** are:

blanked arrow: a graphical symbol depicted as three line segments which form an isosceles triangle. The origin of the symbol corresponds to the intersection point of the two equal sides. The annotation curve, to which the symbol is applied, acts as a bisector to the angle created by the two equal sides. The area within the symbol is blanked.

blanked box: a graphical symbol depicted as four line segments which form a square. Two opposite sides of the square are parallel to the annotation curve to which the symbol is applied. The origin of the symbol is the geometric centre of the square. The area within the symbol is blanked.

blanked dot: a graphical symbol depicted as a circle. The origin of the symbol is the centre of the circle. The area within the symbol is blanked.

dimension origin: a graphical symbol depicted as a circle. The origin of the symbol is the centre of the circle.

filled arrow: a graphical symbol depicted as three line segments which form an isosceles triangle. The origin of the symbol corresponds to the intersection point of the two equal sides. The annotation curve, to which the symbol is applied, acts as a bisector to the angle created by the two equal sides. The area within the symbol is shaded.

filled box: a graphical symbol depicted as four line segments which form a square. Two opposite sides of the square are parallel to the annotation curve to which the symbol is applied. The origin of the symbol is the geometric centre of the square. The area that lies within the symbol is shaded.

filled dot: a graphical symbol depicted as a circle. The origin of the symbol is the centre of the circle. The area within the symbol is shaded.

integral symbol: a graphical symbol depicted as one line segment forming two adjacent arcs. The origin of the symbol is the midpoint between the two arcs.

open arrow: a graphical symbol depicted as three line segments which form an isosceles triangle where the third side of the triangle is blanked. The origin of the symbol corresponds to the intersection point of the two equal sides. The annotation curve, to which the symbol is applied, acts as a bisector to the angle created by the two equal sides.

slash: a graphical symbol depicted as a line segment with the midpoint of the segment being the origin and lying on the annotation curve to which it is applied.

unfilled arrow: a graphical symbol depicted as three line segments which form an isosceles triangle. The origin of the symbol corresponds to the intersection point of the two equal sides. The annotation curve, to which the symbol is applied, acts as a bisector to the angle created by the two equal sides.

5.2.3.1.105 presentation_view_relationship

An **presentation_view_relationship** is a **presentation_representation_relationship** that relates two view areas. See ARM definition for View_area_relationship in paragraph 4.2.493 for more information.

EXPRESS specification:

```
*)
ENTITY presentation_view_relationship
  SUBTYPE OF (representation_relationship);
WHERE
  WR1:  ('AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN
         TYPEOF(SELF\representation_relationship.rep_1))
           AND
        ('AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN
         TYPEOF(SELF\representation_relationship.rep_2));
END_ENTITY;
(*
```

Formal propositions:

WR1: Both **representation**s related with an **presentation_view_relationship** shall be of type **presentation_view**.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **presentation_view_-relationship** entity:

– restrict_multi_language_for_presentation_view_relationship (see 5.2.4.129).

5.2.3.1.106 process_change

A **process_change** is a type of **action_property** that describes the differences between two **action**s concerning the properties of these **action**s.

EXPRESS specification:

```
*)
ENTITY process_change
  SUBTYPE OF (action_property);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.ACTION_RELATIONSHIP' IN TYPEOF(SELF.definition);
```

```
  WR2:  (SIZEOF( USEDIN(SELF, 'AUTOMOTIVE_DESIGN.ACTION_PROPERTY_REPRESENTATION.' +
                             'PROPERTY')) <= 4 ) AND
        (SIZEOF( QUERY( apr <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                              'ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
                 NOT (apr.name IN ['added elements','deleted elements',
                                   'relating change description',
                                   'related change description'])
               )) = 0 );
  WR3:  SIZEOF( QUERY( apr <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                              'ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
                 apr.name = 'added elements'
               )) <= 1;
  WR4:  SIZEOF( QUERY( apr <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                              'ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
                 apr.name = 'deleted elements'
               )) <= 1;
  WR5:  SIZEOF( QUERY( apr <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                              'ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
                 apr.name = 'relating change description'
               )) <= 1;
  WR6:  SIZEOF( QUERY( apr <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                              'ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
                 apr.name = 'related change description'
               )) <= 1;
  WR7:  SIZEOF( QUERY( apr <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                             'ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
                (apr.representation.name = 'relating change descripiton') AND
                (SIZEOF( QUERY( rr <* USEDIN(apr.representation,
                         'AUTOMOTIVE_DESIGN.REPRESENTATION_RELATIONSHIP.REP_1') |
                         (SIZEOF( QUERY( apr_1 <* USEDIN(rr.rep_2,
                                    'AUTOMOTIVE_DESIGN.' +
                                    'ACTION_PROPERTY_REPRESENTATION.' +
                                    'REPRESENTATION') |
                                  apr_1.name = 'related change description'
                                 )) = 1) AND
                         (rr.name = 'change description')
                       )) = 1 )
               )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **process_change** shall reference as its **definition** an **action_relationship.**

WR2: Each **process_change** shall be referenced by at most four **action_property_representation**s. The **name** of this **action_property_representation**s shall be either 'added elements','deleted elements', 'relating change description', or 'related change description'.

WR3: There shall be at most one **action_property_representation** with a **name** of 'added elements' referencing a **process_change.**

WR4: There shall be at most one **action_property_representation** with a **name** of 'deleted elements' referencing a **process_change.**

WR5: There shall be at exactly one **action_property_representation** with a **name** of 'relating change description' referencing a **process_change.**

WR6: There shall be at exactly one **action_property_representation** with a **name** of 'related change description' referencing a **process_change.**

WR7: The representation, that is referenced as representation by the **action_property_representation** with a **name** of 'relating change description' shall play the role of **rep_1** in a **representation_relationship** where the role of **rep_2** is played by a representation that is referenced as representation by the **action_property_representation** with a **name** of 'related change description'. This **representation_relationship** shall have a **name** of 'change description'.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **process_change** entity:

– restrict_multi_language_for_process_change (see 5.2.4.130).

5.2.3.1.107 process_plan

A **process_plan** is a type of **product_definition_process** that specifies the information necessary to realize or produce a product. See ARM definition for Process_plan in paragraph 4.2.341 for more information.

EXPRESS specification:

```
*)
ENTITY process_plan
  SUBTYPE OF (product_definition_process);
WHERE
  WR1:  SIZEOF( QUERY( ppa <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                              'PROCESS_PRODUCT_ASSOCIATION.PROCESS') |
                (ppa.name <> 'product output') OR
                NOT ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN
                     TYPEOF(ppa.defined_product))
              )) = 0;
  WR2:  SIZEOF( QUERY( ar <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                             'ACTION_RELATIONSHIP.RELATED_ACTION') |
                NOT ('AUTOMOTIVE_DESIGN.PROCESS_PLAN' IN
                     TYPEOF(ar.relating_action))
              )) = 0;
  WR3:  SIZEOF( QUERY( ar <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                             'ACTION_RELATIONSHIP.RELATING_ACTION') |
                NOT ('AUTOMOTIVE_DESIGN.PROCESS_PLAN' IN
                     TYPEOF(ar.related_action)) AND
                ((ar.name <> 'process plan operation assignment') OR
                 (ar.related_action.description <> 'process operation'))
              )) = 0;
```

```
END_ENTITY;
(*
```

Formal propositions:

WR1: The **process_plan** shall only be referenced as **process**by **process_product_associations** that have a **name** of 'produced output' and that reference as **defined_product** a **product_definition.**

WR2: The **process_plan** shall only be referenced as the **related_action** by **action_relationships** that reference as the **relating_action** a **process_plan.**

WR3: The **process_plan** shall only be referenced as the **relating_action** by **action_relationships** that reference as the **related_action** a **process_plan,** or by **action_relationships** that have a **name** of 'process plan operation assignment' and that reference as the **related_action** an **action** that has a **description** of 'process operation'.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **process_plan** entity:

– restrict_multi_language_for_process_plan (see 5.2.4.131);

– restrict_version_assignment_for_process_plan (see 5.2.4.175).

5.2.3.1.108 process_plan_assignment

A **process_plan_assignment** is a type of **action_assignment** that assigns a **process_plan** to a **process_plan_item** that defines the input or output of a process operation within the context of the assigned **process_plan.**

EXPRESS specification:

```
*)
ENTITY process_plan_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF process_plan_item;
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.PROCESS_PLAN' IN TYPEOF(SELF.assigned_action);
  WR2:  Self.role.name = 'process plan context';
  WR3:  SIZEOF( QUERY( ar <* USEDIN( SELF.assigned_action, 'AUTOMOTIVE_DESIGN.' +
                            'ACTION_RELATIONSHIP.RELATING_ACTION') |
                (ar.name = 'process plan operation assignment') AND
                (ar.related_action.description = 'process operation') AND
                (SIZEOF( QUERY( ppa <* USEDIN( ar.related_action,
                                'AUTOMOTIVE_DESIGN.' +
                                'PROCESS_PRODUCT_ASSOCIATION.PROCESS') |
                         SIZEOF( QUERY( i <* SELF.items |
                                 i :=: ppa )) = 1
                       )) = 1 )
              )) = 1;
```

```
  WR4:  (SIZEOF( QUERY( i <* SELF.items |
                 NOT ('AUTOMOTIVE_DESIGN.PROCESS_PRODUCT_ASSOCIATION' IN
                      TYPEOF(i)) )) = SIZEOF(items)) OR
        (SIZEOF( QUERY( i <* SELF.items |
                 ('AUTOMOTIVE_DESIGN.PROCESS_PRODUCT_ASSOCIATION' IN TYPEOF(i)) AND
                 (i\process_product_association.process.description =
                    'process operation') AND
                 (SIZEOF( QUERY( ar <* USEDIN
                                       (i, 'AUTOMOTIVE_DESIGN.ACTION_RELATIONSHIP.' +
                                           'RELATED_ACTION') |
                           (ar.name = 'process plan operation assignment') AND
                           (ar.relating_action :=: SELF.assigned_action)
                         )) = 1 )
               )) >= 1);
END_ENTITY;
(*
```

Attribute definitions:

items: The set of items to which the **process_plan** is assigned.

Formal propositions:

WR1: The **process_plan_assignment** shall reference as its **assigned_action** an **action** that is of type **process_plan.**

WR2: The **process_plan_assignment** shall reference as its **role** an **action_role** that has a **name** of 'process plan context'.

WR3: The **process_plan** that is referenced as the **assigned_action** by the **process_plan_-assignment** shall be referenced as the **relating_action** by an **action_relationship** that has a **name** of 'process plan operation assignment' and that references as the **related_action** an **action** that has a **description** of 'process operation' and that is referenced as the **process** by a **process_product_association** that is contained in the set of **items** of the **process_plan_-assignment.**

WR4: The **process_plan_assignment** shall only contain **process_product_association**s in its set of **items** that reference as their **process** an **action** that has a **description** of 'process operation' and that is referenced as the **related_action** by an **action_relationship** that has a **name** of 'process plan operation assignment' and that references as the **relating_action** the **process_plan** that is referenced as the **assigned_action** by the **process_plan_assignment.**

5.2.3.1.109 product_class

A **product_class** is a type of **product_concept** and of **characterized_object** that is the identification of a set of similar products to be offered to the market. See ARM definition for Product_class in paragraph 4.2.348 for more information.

EXPRESS specification:

```
*)
ENTITY product_class
  SUBTYPE OF (characterized_object, product_concept);
WHERE
  WR1:  SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'CONFIGURATION_ITEM.ITEM_CONCEPT')) =
        SIZEOF( QUERY ( ci <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'CONFIGURATION_ITEM.ITEM_CONCEPT') |
                'AUTOMOTIVE_DESIGN.PRODUCT_COMPONENT' IN
                 TYPEOF(ci))) +
        SIZEOF( QUERY( ci <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'CONFIGURATION_ITEM.ITEM_CONCEPT') |
                ('AUTOMOTIVE_DESIGN.CONFIGURABLE_ITEM' IN
                 TYPEOF(ci)) AND
                (ci.description = 'class function definition') AND
                (SIZEOF( QUERY( cd <* USEDIN(ci, 'AUTOMOTIVE_DESIGN.' +
                                     'CONFIGURATION_DESIGN.CONFIGURATION') |
                         ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN
                           TYPEOF(cd.design)) AND
                         (cd.design\product_definition.frame_of_reference.name =
                          'functional_definition')
                       )) > 0 ) )) +
        SIZEOF( QUERY( ci <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'CONFIGURATION_ITEM.ITEM_CONCEPT') |
                ci.description = 'product specification'));
  WR2:  SIZEOF( QUERY( ci <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                       'CONFIGURATION_ITEM.ITEM_CONCEPT') |
                'AUTOMOTIVE_DESIGN.PRODUCT_COMPONENT' IN
                 TYPEOF(ci))) <= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **product_class** shall be referenced as the **item_concept** only by **configuration_items** that are of type **product_component**, by **configuration_items** that have a **description** of 'class function definition' and that are referenced as the **configuration** by at least one **configuration_design** that references as its **design** a **product_definition** that references as its **frame_of_reference** a **product_definition_context** with a **name** of 'functional definition', or by **configuration_items** that are of type **configurable_item** and have a **description** of 'product specification'.

WR2: The **product_class** shall be referenced as the **item_concept** by at most one **configuration_item** that is of type **product_component.**

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_class** entity:

– restrict_multi_language_for_product_class (see 5.2.4.135);

– restrict_version_assignment_for_product_class (see 5.2.4.176).

5.2.3.1.110 product_component

A **product_component** is a type of **configuration_item_with_context** and of **characterized_object** that identifies an element in the decomposition structure of the products belonging to a **product_class.** See ARM definition for Product_component in paragraph 4.2.350 for more information.

EXPRESS specification:

```
*)
ENTITY product_component
  SUBTYPE OF (characterized_object, configuration_item_with_context);
WHERE
  WR1:  SELF.purpose IN ['instance required','no instance required'];
END_ENTITY;
(*
```

Formal propositions:

WR1: The **product_component** shall have a **purpose** of either 'instance required' or 'no instance required'.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_component** entity:

– restrict_multi_language_for_product_component (see 5.2.4.136);

– restrict_version_assignment_for_product_component (see 5.2.4.177).

5.2.3.1.111 product_concept_feature_category

A **product_concept_feature_category** is a type of group that specifies a set of similar **product_concept_feature** instances. See ARM definition for Specification_category in paragraph 4.2.416 for more information.

EXPRESS specification:

```
*)
ENTITY product_concept_feature_category
  SUBTYPE OF (group);
WHERE
  WR1:  SIZEOF( QUERY( gma <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                              'GROUP_ASSIGNMENT.ASSIGNED_GROUP') |
                ('AUTOMOTIVE_DESIGN.GROUP_MEMBER_ASSIGNMENT' IN
                  TYPEOF(gma)) AND
```

```
                (SIZEOF( QUERY( gma_i <* gma.items |
                         'AUTOMOTIVE_DESIGN.CONDITIONAL_CONCEPT_FEATURE' IN
                          TYPEOF(gma_i)
                       )) > 0 )
        )) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **product_concept_feature_category** shall be the **assigned_group** of a **group_member_assignment** that contains only **product_concept_feature**s in its set of **items** that are not of type **conditional_concept_feature.**

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **product_concept_feature_category** entity:

– restrict_multi_language_for_product_concept_feature_category (see 5.2.4.139).

5.2.3.1.112 product_concept_feature_category_usage

A **product_concept_feature_category_usage** is a type of **group_assignment** that assigns a **product_concept_feature_category** to a **product_class.** The **product_concept_feature_category_usage** specifies if the usage of one or more of the grouped **product_concept_feature** instances is mandatory or optional for all products of that **product_class.** See ARM definition for Class_category_association in paragraph 4.2.64 for more information.

EXPRESS specification:

```
*)
ENTITY product_concept_feature_category_usage
  SUBTYPE OF (group_assignment);
  items : SET [1:?] OF category_usage_item;
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_FEATURE_CATEGORY' IN
         TYPEOF(SELF.assigned_group);
  WR2:  SELF.role.name IN ['mandatory category usega','optional category usage'];
END_ENTITY;
(*
```

Attribute definitions:

items: The set of items to which the **product_concept_feature_category** is assigned.

Formal propositions:

WR1: The **product_concept_feature_category_usage** shall reference as its **assigned_group** a **group** that is of type **product_concept_feature_category.**

WR2: The **product_concept_feature_category_usage** shall reference as its **role** a **group_role** that has a **name** of either 'mandatory category usage' or 'optional category usage'.

5.2.3.1.113 product_definition_occurrence

A **product_definition_occurrence** is a type of **product_definition** that identifies an occurrence of an object that is defined by either a **product_definition** that acts as a definitional template for a product, or by a **configurable_item** that identifies a physically realizable object by selecting a set of **product_concept_feature** instances for a product class. See ARM definition for Item_instance in paragraph 4.2.241, for Selected_instance in paragraph 4.2.392, for Single_instance in paragraph 4.2.404, and for Quantified_instance in paragraph 4.2.365 for more information.

EXPRESS specification:

```
*)
ENTITY product_definition_occurrence
  SUBTYPE OF (product_definition);
WHERE
  WR1:  SELF.name IN ['quantified instance','selected instance','single instance'];
  WR2:  SELF.frame_of_reference.name = 'definitional template';
  WR3:  (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                        'PRODUCT_DEFINITION_RELATIONSHIP.' +
                                        'RELATED_PRODUCT_DEFINITION') |
                 'AUTOMOTIVE_DESIGN.ASSEMBLY_COMPONENT_USAGE' IN
                  TYPEOF(pdr))) > 0 ) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                               'PRODUCT_DEFINITION_RELATIONSHIP.' +
                               'RELATED_PRODUCT_DEFINITION') |
                 (pdr.name = 'solution instance association') AND
                 ('AUTOMOTIVE_DESIGN.SOLUTION_BASED_PRODUCT_DEFINITION' IN
                  TYPEOF(pdr.relating_product_definition))
               )) > 0 );
  WR4:  (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                               'PRODUCT_DEFINITION_RELATIONSHIP.' +
                               'RELATED_PRODUCT_DEFINITION') |
                 (pdr.name = 'definition usage') AND
                 (pdr.relating_product_definition.frame_of_reference.name =
                  'definitional template')
               )) = 1 ) OR
        (SIZEOF( QUERY( cd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                      'CONFIGURATION_DESIGN.DESIGN') |
                 'AUTOMOTIVE_DESIGN.OCCURRENCE_USAGE_DEFINITION' IN
                  TYPEOF(cd))) = 1 );
  WR5:  (SELF.name <> 'quantified instance') OR
```

```
      ((SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                           'PROPERTY_DEFINITION.DEFINITION') |
                 pd.description = 'occurrence quantity'
               )) = 1) AND
       (SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                           'PROPERTY_DEFINITION.DEFINITION') |
                 (pd.description = 'occurrence quantity') AND
                 (SIZEOF( USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                          'PROPERTY_DEFINITION_REPRESENTATION.' +
                          'DEFINITION')) = 1 ) AND
                 (SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                          (pdr.used_representation.description = 'quantity') AND
                          (SIZEOF(pdr.used_representation.items) = 1) AND
                          (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                                   ('AUTOMOTIVE_DESIGN.' +
                                    'MEASURE_REPRESENTATION_ITEM' IN
                                    TYPEOF(r_i)) AND
                                   (r_i.name = 'quantity measure') )) = 1 )
                       )) = 1 )
               )) = 1));
WR6:  (SELF.name <> 'selected instance') OR
      ((SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                           'PROPERTY_DEFINITION.DEFINITION') |
                 pd.description = 'occurrence quantity'
               )) = 1) AND
       (SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                           'PROPERTY_DEFINITION.DEFINITION') |
                 (pd.description = 'occurrence quantity') AND
                 (SIZEOF( USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                          'PROPERTY_DEFINITION_REPRESENTATION.' +
                          'DEFINITION')) = 1 ) AND
                 (SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
                                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                          (pdr.used_representation.description =
                           'selection control') AND
                          ({1 <= SIZEOF(pdr.used_representation.items) <= 3}) AND
                          (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                                  ('AUTOMOTIVE_DESIGN.' +
                                   'DESCRIPTIVE_REPRESENTATION_ITEM' IN
                                   TYPEOF(r_i)) AND
                                  (r_i.name = 'selection condition') )) = 1 ) AND
                          (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                                  ('AUTOMOTIVE_DESIGN.' +
                                   'MEASURE_REPRESENTATION_ITEM' IN
                                   TYPEOF(r_i)) AND
                                  (r_i.name = 'maximum quantity') )) <= 1 ) AND
                          (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                                  ('AUTOMOTIVE_DESIGN.' +
                                   'MEASURE_REPRESENTATION_ITEM' IN
                                   TYPEOF(r_i)) AND
                                  (r_i.name = 'minimum quantity') )) <= 1 )
```

```
                        )) = 1 )
                )) = 1));
END_ENTITY;
(*
```

Formal propositions:

WR1: The **product_definition_occurrence** shall have a **name** of 'quantified instance', 'selected instance', or 'single instance'.

WR2: The **product_definition_occurrence** shall reference as its **frame_of_reference** a **product_definition_context** with a **name** of 'definitional template'.

WR3: The **product_definition_occurrence** shall be referenced as the **related_product_definition** by at least one **assembly_component_usage** or shall be referenced as the **related_product_definition** by at least one **product_definition_relationship** that has a **name** of 'solution instance association' and that references as its **relating_product_definition** a **solution_based_product_definition.**

WR4: The **product_definition_occurrence** shall either be referenced as the **related_product_definition** by exactly one **product_definition_relationship** that has a **name** of 'definition usage' and that references as the **relating_product_definition** a **product_definition** that references as its **frame_of_reference** a **product_definition_context** that has a **name** of 'definitional template', or shall be referenced as the **design** by exactly one **configuration_design** that is of type **occurrence_usage_definition.**

WR5: If the **product_definition_occurrence** has a **name** of 'quantified instance', it shall be referenced as the **definition** by exactly one **property_definition** with a **description** of 'occurrence quantity'. This **property_definition** shall be referenced as the **definition** by exactly one **property_definition_representation.** This **property_definition_representation** shall reference as its **used_representation** a **representation** that has a **description** of 'quantity' and that contains exactly one **representation_item** in its set of **items**. This **representation_item** shall be a **measure_representation_item** and shall have a **name** of 'quantity measure'.

WR6: If the **product_definition_occurrence** has a **name** of 'selected instance', it shall be referenced as the **definition** by exactly one **property_definition** with a **description** of 'occurrence quantity'. This **property_definition** shall be referenced as the **definition** by exactly one **property_definition_representation.** This **property_definition_representation** shall reference as its **used_representation** a **representation** that has a **description** of 'selection control' and that contains between one and three **representation_items** in its set of **items**. These **representation_items** shall either be a **descriptive_representation_item** with a **name** of 'selection condition' or shall be a **measure_representation_item** with a **name** of either 'maximum quantity' or 'minimum quantity'. Exactly one of these **representation_items** shall be a **descriptive_representation_item** and shall have a **name** of 'selection condition'. At most one shall be a **measure_representation_item** and shall have a **name** of 'maximum quantity'. At most one shall be a **measure_representation_item** and shall have a **name** of 'minimum quantity'.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **product_definition_-occurrence** entity:

– restrict_multi_language_for_product_definition_occurrence (see 5.2.4.143).

5.2.3.1.114 product_definition_resource

A **product_definition_resource** is a type of **action_resource** and of **product_definition** that is used as a resource to perform a process operation and that has characteristics of a product.

EXPRESS specification:

```
*)
ENTITY product_definition_resource
  SUBTYPE OF (action_resource, product_definition);
WHERE
  WR1:  SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES')) >= 1;
  WR2:  SIZEOF( QUERY( rfar <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                 'REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES') |
                NOT (rfar.name IN ['reference tool','non reference toll']) OR
                (SIZEOF( QUERY( a <* rfar.operations |
                         a.name <> 'process operation'
                       )) > 0 )
             )) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **product_definition_resource** shall be contained in the set of **resources** of at least one **requirement_for_action_resource.**

WR2: The **product_definition_resource** shall be contained in the set of **resources** of only **requirement_for_action_resource** instances that have a name of either 'reference tool' or 'non reference tool' and that contain only **action**s in their set of **operations** that have a **description** of 'process operation'.

5.2.3.1.115 project

A **project** is a type of organization representing a work programme that consists of a set of assigned **action**s. See ARM definition for Project in paragraph 4.2.356 for more information.

EXPRESS specification:

```
*)
ENTITY project
```

```
  SUBTYPE OF (organization);
WHERE
  WR1:  SIZEOF( QUERY( ada <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
                'APPLIED_DATE_ASSIGNMENT.ITEMS') |
         ada.role.name = 'actual start' )) +
        SIZEOF( QUERY( adata <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
         adata.role.name = 'actual start' )) <= 1;
  WR2:  SIZEOF( QUERY( ada <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
                'APPLIED_DATE_ASSIGNMENT.ITEMS') |
         ada.role.name = 'actual end' )) +
        SIZEOF( QUERY( adata <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
         adata.role.name = 'actual end' )) <= 1;
  WR3:  SIZEOF( QUERY( ada <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
                'APPLIED_DATE_ASSIGNMENT.ITEMS') |
         ada.role.name = 'planned end' )) +
        SIZEOF( QUERY( adata <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
         adata.role.name = 'planned end' )) +
        SIZEOF( QUERY( ea <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
               'EVENT_ASSIGNMENT.ITEMS') |
         ea.role.name = 'planned end' )) <= 1;
  WR4:  SIZEOF( QUERY( ada <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
                'APPLIED_DATE_ASSIGNMENT.ITEMS') |
         ada.role.name = 'actual start' )) +
        SIZEOF( QUERY( adata <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
         adata.role.name = 'actual start' )) <= 1;
  WR5:  SIZEOF( QUERY( ea <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
               'EVENT_ASSIGNMENT.ITEMS') |
         ea.role.name = 'planned start' )) <= 1;
  WR6:  (SIZEOF( QUERY( ea <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
                'EVENT_ASSIGNMENT.ITEMS') |
          ea.role.name = 'actual start' )) <= 1 ) AND
        ((SIZEOF( QUERY( ea <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
                 'EVENT_ASSIGNMENT.ITEMS') |
           ea.role.name = 'actual start' )) = 0 ) OR
         (SIZEOF( QUERY( ea <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
                 'EVENT_ASSIGNMENT.ITEMS') |
           (ea.role.name = 'actual start' ) AND
           ('AUTOMOTIVE_DESIGN.EVENT_REFERENCE' IN TYPEOF(ea.assigned_action)) AND
           (SIZEOF( QUERY( ar <* USEDIN( ea, 'AUTOMOTIVE_DESIGN.' +
         'ACTION_RELATIONSHIP.RELATING_ACTION') |
            (ar.name = 'event relationship') AND
            ('AUTOMOTIVE_DESIGN.EVENT_REFERENCE' IN
             TYPEOF(ar.related_action)) AND
            (SIZEOF( QUERY( aa <* USEDIN(ar.related_action,'AUTOMOTIVE_DESIGN.' +
          'ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
             'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT' IN
              TYPEOF(aa))) > 0 ) AND
            (SIZEOF( QUERY( ea1 <* USEDIN(ar.related_action,
```

```
          'AUTOMOTIVE_DESIGN.ACTION_ASSIGNMENT.' +
          'ASSIGNED_ACTION') |
            ('AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT' IN
             TYPEOF(ea1)) AND
            (ea1.role.name = 'planned end') AND
            (SIZEOF( QUERY( ea1_i <* ea1\event_assignment.items |
              SELF :=: ea1_i )) = 1 )
          )) > 0 )
         )) > 0 )
        )) = 1 ));
  WR7:  SIZEOF( QUERY( aaa <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
             'APPLIED_ACTION_ASSIGNMENT.ITEMS') |
        aaa.role.name <> 'work program' )) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **project** shall either be contained in the set of **items** of at most one **applied_date_assignment** that references as its **role** a **date_role** with a **name** of 'actual start', or it shall be contained in the set of **items** of at most one **applied_date_and_time_assignment** that references as its **role** a **date_time_role** with a **name** of 'actual start'.

WR2: The **project** shall either be contained in the set of **items** of at most one **applied_date_assignment** that references as its **role** a **date_role** with a **name** of 'actual end', or it shall be contained in the set of **items** of at most one **applied_date_and_time_assignment** that references as its **role** a **date_time_role** with a **name** of 'actual end'.

WR3: The **project** shall either be contained in the set of **items** of at most one **applied_date_assignment** that references as its **role** a **date_role** with a **name** of 'planned end', or shall be contained in the set of **items** of at most one **applied_date_and_time_assignment** that references as its **role** a **date_time_role** with a **name** of 'planned end', or shall be be contained in the set of **items** of at most one **event_assignment** that references as its **role** an **action_role** with a **name** of 'planned end'.

WR4: The **project** shall either be contained in the set of **items** of at most one **applied_date_assignment** that references as its **role** a **date_role** with a **name** of 'planned start', or shall be contained in the set of **items** of at most one **applied_date_and_time_assignment** that references as its **role** a **date_time_role** with a **name** of 'planned start'.

WR5: The **project** shall be contained in the set of **items** of at most one **event_assignment** that references as its **role** an **action_role** with a **name** of 'planned start'.

WR6: The **project** shall be contained in the set of **items** of at most one **event_assignment** that references as its **role** an **action_role** with a **name** of 'actual start'. If present, this **event_assignment** shall reference as its **assigned_action** an **event_reference** that is the **relating_action** in an **action_relationship** that has a **name** of 'event relationship' and shall reference as its **related_action** an **event_reference** that is referenced as the **assigned_action** by an **event_assignment** that references as its **role** an **action_role** with a **name** of 'planned end' and that contains the **project** in its set of **items**.

WR7: The **project** shall only be contained in the set of **items** of **applied_action_assignment**s that reference as their **role** an **action_role** with a **name** of 'work program'.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **project** entity:

– restrict_multi_language_for_project (see 5.2.4.147).

5.2.3.1.116 project_assignment

A **project_assignment** is an **organization_assignment** that assigns a **project** to a **product_class** that is affected by the **project**.

EXPRESS specification:

```
*)
ENTITY project_assignment
  SUBTYPE OF (organization_assignment);
  items : SET [1:?] OF project_item;
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.PROJECT' IN TYPEOF(SELF.assigned_organization);
  WR2:  SELF.role.name = 'affecting project';
END_ENTITY;
(*
```

Attribute definitions:

items: The set of **product_class** instances that are affected by the **project**.

Formal propositions:

WR1: The **project_assignment** shall reference as its **assigned_organization** an **organization** that is of type **project**.

WR2: The **project_assignment** shall reference as its **role** an **organization_role** that has a **name** of 'affecting project'.

5.2.3.1.117 property_change

A **property_change** is a type of **property_definition** that describes the differences between two objects concerning the properties of these objects.

EXPRESS specification:

```
*)
ENTITY property_change
  SUBTYPE OF (property_definition);
WHERE
```

```
  WR1:  SIZEOF(['AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RELATIONSHIP',
                'AUTOMOTIVE_DESIGN.SHAPE_ASPECT_RELATIONSHIP']
               * TYPEOF(SELF.definition)) = 1;
  WR2:  (SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                 'PROPERTY_DEFINITION_RELATIONSHIP.' +
                 'RELATING_PROPERTY_DEFINITION')) <= 2) AND
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_RELATIONSHIP.' +
                                'RELATING_PROPERTY_DEFINITION') |
                 NOT (pdr.name IN ['added property','deleted property'])
               )) = 0);
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION_RELATIONSHIP.' +
                              'RELATING_PROPERTY_DEFINITION') |
                pdr.name = 'added property'
               )) <= 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION_RELATIONSHIP.' +
                              'RELATING_PROPERTY_DEFINITION') |
                pdr.name = 'deleted property'
               )) <= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **property_change** shall reference as its **definition** either a **product_definition_relationship** or a **shape_aspect_relationship.**

WR2: Each **property_change** shall play the role of **relating_property_definition** for at most two **property_definition_relationship**s. The **name** of this **property_definition_relationship**s shall be either 'added property' or 'deleted property'.

WR3: There shall be at most one **property_definition_relationship** with a **name** of 'added property' associated with a **property_change.**

WR4: There shall be at most one **property_definition_relationship** with a **name** of 'deleted property' associated with a **property_change.**

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **property_change** entity:

– restrict_multi_language_for_property_change (see 5.2.4.148).

5.2.3.1.118 property_value_representation

A **property_value_representation** is a type of **representation** that represents a value of a property. See ARM definition for Property_value in paragraph 4.2.364 for more information.

EXPRESS specification:

```
*)
ENTITY property_value_representation
  SUBTYPE OF (representation);
WHERE
  WR1:  SIZEOF(SELF.items) <= 3;
  WR2:  SIZEOF( QUERY( i <* SELF.items |
                SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM']
                       * TYPEOF(i)) = 0
              )) = 0;
  WR3:  SIZEOF( QUERY( i <* SELF.items |
                i.name = 'lower'
              )) <= 1;
  WR4:  SIZEOF( QUERY( i <* SELF.items |
                i.name = 'upper'
              )) <= 1;
  WR5:  SIZEOF( QUERY( i <* SELF.items |
                i.name = 'specified value'
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: Each **property_value_representation** shall contain at most three **representation_items** in its set of **items**.

WR2: Each **representation_item** in the set of **items** of a **property_value_representation** shall be either of type **measure_representation_item** or **descriptive_representation_item.**

WR3: There shall be at most one **representation_item** with a **name** of 'lower' in the set of **items** of a **property_value_representation.**

WR4: There shall be at most one **representation_item** with a **name** of 'upper' in the set of **items** of a **property_value_representation.**

WR5: There shall be exactly one **representation_item** with a **name** of 'specified value' in the set of **items** of a **property_value_representation.**

5.2.3.1.119 rectangular_closed_profile

A **rectangular_closed_profile** is a type of **shape_aspect** that consists of an enclosed 2D area and a reference system for the parameters of the **rectangular_closed_profile.** The enclosed area is defined by four straight sides with opposite sides equal in length. The reference system is used to define location and orientation of the **rectangular_closed_profile** and consists of two orthogonal reference axes X and Y that intersect in a reference point. See ARM definition for Rectangular_closed_profile in paragraph 4.2.370 for more information.

EXPRESS specification:

```
*)
ENTITY rectangular_closed_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)
              )) = 1;
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
         ({3 <= SIZEOF(pdr.used_representation.items) <= 4})
              )) <= 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
         (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         NOT (srwp_i.name IN ['orientation','length',
                                              'width','corner radius'])
                       )) > 0 )
              )) = 0;
  WR5:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'orientation') AND
                         ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i))
                       )) = 1)
              )) = 1;
  WR6:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'width') AND
                         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                  'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                  * TYPEOF(srwp_i)) = 2 )
                       )) = 1)
              )) = 1;
  WR7:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
```

```
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'length') AND
                         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                  'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                  * TYPEOF(srwp_i)) = 2 )
                       )) = 1)
             )) = 1;
  WR8:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'corner radius') AND
                         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                  'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                  * TYPEOF(srwp_i)) = 2 )
                       )) = 1)
             )) <= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **rectangular_closed_profile** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **rectangular_closed_profile** shall have exactly one implicit representation defined by a **shape_representation_with_parameters.**

WR3: The **shape_representation_with_parameters** that represents the **rectangular_closed_profile** shall contain three or four **representation_items** in its set of **items**.

WR4: The implicit representation of a **rectangular_closed_profile** shall contain only **representation_items** in its set of **items** that have a **name** of either 'orientation', 'length', 'width', or 'corner radius'.

WR5: Exactly one **representation_item** used for the implicit representation of a **rectangular_closed_profile** shall be of type **placement** with a **name** of 'orientation'.

WR6: Exactly one **representation_item** used for the implicit representation of a **rectangular_closed_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'width'.

WR7: Exactly one **representation_item** used for the implicit representation of a **rectangular_closed_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'length'.

WR8: At most one **representation_item** used for the implicit representation of a **rectangular_closed_profile** shall have a **name** of 'corner radius'. If present, this **representation_item** shall be of type **measure_representation_item** and **length_measure_with_unit.**

Informal propositions:

IP1: The **rectangular_closed_profile** shall be planar and shall lie in the X-Y plane of the reference system.

IP2: The reference point of the **rectangular_closed_profile** is defined at the centre of the rectangle.

IP3: The length of the **rectangular_closed_profile** is defined in the X direction and the width is defined in the Y direction of the reference system.

5.2.3.1.120 rectangular_pattern

A **rectangular_pattern** is a type of **replicate_feature** that relates a base feature and one or more **shape_aspects** which are the placement of the base feature at a specified row and column location on the base part. A position is defined on the part and a number of base features are equally place in rows and columns. See ARM definition for Rectangular_pattern in paragraph 4.2.374 for more information.

EXPRESS specification:

```
*)
ENTITY rectangular_pattern
  SUBTYPE OF (replicate_feature);
WHERE
  WR1:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN
                   TYPEOF(pdr.used_representation)) AND
         (pdr.used_representation.description = 'row layout direction')
               )) = 1
              )) = 1;
  WR2:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN
           TYPEOF(pdr.used_representation)) AND
         (pdr.used_representation.description = 'column layout direction')
               )) = 1
              )) = 1;
  WR3:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
         (SIZEOF(pdr.used_representation.items) = 5 )
```

```
            )) = 1
           )) = 1;
WR4:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
           'PROPERTY_DEFINITION.DEFINITION') |
       SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
         ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN
           TYPEOF(srwp_i)) AND
         ('AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN
           TYPEOF(srwp_i\measure_representation_item.value_component)) AND
         (srwp_i.name = 'number of rows') )) = 1 )
            )) = 1
           )) = 1;
WR5:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
           'PROPERTY_DEFINITION.DEFINITION') |
       SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
       (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
        ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN
          TYPEOF(srwp_i)) AND
        ('AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN
          TYPEOF(srwp_i\measure_representation_item.value_component)) AND
        (srwp_i.name = 'number of columns') )) = 1 )
            )) = 1
           )) = 1;
WR6:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
           'PROPERTY_DEFINITION.DEFINITION') |
       SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
       (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
        (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(srwp_i)) = 2 ) AND
        (srwp_i.name = 'row spacing') )) = 1 )
            )) = 1
           )) = 1;
WR7:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
           'PROPERTY_DEFINITION.DEFINITION') |
       SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
       (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
        (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
```

```
          * TYPEOF(srwp_i)) = 2 ) AND
         (srwp_i.name = 'column spacing') )) = 1 )
             )) = 1
            )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **rectangular_pattern** shall have exactly one **direction_shape_representation** with a **description** of 'row layout direction'.

WR2: The **rectangular_pattern** shall have exactly one **direction_shape_representation** with a **description** of 'column layout direction'.

WR3: The implicit representation of the **rectangular_pattern** shall contain five **representation_items** in its set of **items**.

WR4: The implicit representation of the **rectangular_pattern** shall contain exactly one **representation_item** of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'number of rows'.

WR5: The implicit representation of the **rectangular_pattern** shall contain exactly one **representation_item** of type **measure_representation_item** with a **value_component** of type **count_measure** and a **name** of 'number of columns'.

WR6: The implicit representation of the **rectangular_pattern** shall contain exactly one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'row spacing'.

WR7: The implicit representation of the **rectangular_pattern** shall contain exactly one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'column spacing'.

Informal propositions:

IP1: The location of the **rectangular_pattern** shall be defined at the position of the first base feature.

IP2: The **rectangular_pattern** shall be defined in the X-Y plane.

5.2.3.1.121 rep_item_group

A **rep_item_group** is a type of **group** and **representation_item** that may be part of a layer or a template. See ARM definition for Group in paragraph 4.2.222 and Structuring_method_hierarchy in paragraph 4.2.426 for more information.

EXPRESS specification:

```
*)
```

```
ENTITY rep_item_group
  SUBTYPE OF (group, representation_item);
WHERE
  WR1:  SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PRESENTATION_LAYER_ASSIGNMENT.ASSIGNED_ITEMS')) > 0;
  WR2:  SIZEOF( QUERY( gma <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                               'GROUP_ASSIGNMENT.ASSIGNED_GROUP') |
                ('AUTOMOTIVE_DESIGN.GROUP_MEMBER_ASSIGNMENT' IN
                  TYPEOF(gma)) AND
                (SIZEOF( QUERY( gma_i <* gma.items |
                         NOT ('AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_ITEM' IN
                               TYPEOF(gma_i))
                       )) > 0 )
              )) = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: A **rep_item_group** shall be contained in the set of **assigned_items** of a **presentation_layer_assignment.**

WR2: The items in a **rep_item_group** shall be of type **geometric_representation_item.**

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **rep_item_group** entity:

– restrict_multi_language_for_rep_item_group (see 5.2.4.152).

5.2.3.1.122 replicate_feature

A **replicate_feature** is a type of **feature_definition** that is a **feature_definition** for the purpose of multiple representations. See ARM definition for Replicate_feature in paragraph 4.2.380 for more information.

EXPRESS specification:

```
*)
ENTITY replicate_feature
  SUBTYPE OF (feature_definition);
WHERE
  WR1:  SIZEOF( QUERY( cor <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'CHARACTERIZED_OBJECT_RELATIONSHIP.RELATED_OBJECT') |
          (cor.name = 'replicate base assignment') AND
          ('AUTOMOTIVE_DESIGN.FEATURE_DEFINITION' IN
            TYPEOF(cor.relating_object))
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **replicate_feature** shall be the **related_object** in exactly one **characterized_object_relationship** with a **name** of 'replicate base assignment' in which the **relating_object** is a **feature_definition.**

5.2.3.1.123 retention

A **retention** is a type of **action** that defines a period of time that product data needs to be retained due to organizational policy or to legal requirements. See See ARM definition for Retention_period in paragraph 4.2.381 for more information.

EXPRESS specification:

```
*)
ENTITY retention
  SUBTYPE OF (action);
WHERE
  WR1:  SIZEOF( QUERY( aa <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
               'AUTOMOTIVE_DESIGN.RETENTION_ASSIGNMENT' IN
                TYPEOF(aa))) > 0;
  WR2:  SIZEOF( QUERY( ada <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
               'APPLIED_DATE_ASSIGNMENT.ITEMS') |
         ada.role.name = 'start' )) +
        SIZEOF( QUERY( adata <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
         adata.role.name = 'start' )) +
        SIZEOF( QUERY( ar <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
              'ACTION_RELATIONSHIP.RELATING_ACTION') |
         ar.name = 'start' )) = 1;
  WR3:  SIZEOF( QUERY( ada <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
               'APPLIED_DATE_ASSIGNMENT.ITEMS') |
         ada.role.name = 'earliest end' )) +
        SIZEOF( QUERY( adata <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
         adata.role.name = 'earliest end' )) +
        SIZEOF( QUERY( ar <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
              'ACTION_RELATIONSHIP.RELATING_ACTION') |
         ar.name = 'earliest end' )) +
        SIZEOF( QUERY( ap <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
              'ACTION_PROPERTY.DEFINITION') |
         (ap.name = 'retention period' ) AND
         (SIZEOF( QUERY( apr <* USEDIN( ap, 'AUTOMOTIVE_DESIGN.' +
                'ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
          SIZEOF( QUERY( apr_r_i <* apr.representation.items |
          apr_r_i.name = 'earliest end' )) > 0
                )) > 0 ) )) = 1;
  WR4:  SIZEOF( QUERY( ada <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
               'APPLIED_DATE_ASSIGNMENT.ITEMS') |
```

```
        ada.role.name = 'latest end' )) +
       SIZEOF( QUERY( adata <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
        adata.role.name = 'latest end' )) +
       SIZEOF( QUERY( ar <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
             'ACTION_RELATIONSHIP.RELATING_ACTION') |
        ar.name = 'latest end' )) +
       SIZEOF( QUERY( ap <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
             'ACTION_PROPERTY.DEFINITION') |
        (ap.name = 'retention period' ) AND
        (SIZEOF( QUERY( apr <* USEDIN( ap, 'AUTOMOTIVE_DESIGN.' +
               'ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
         SIZEOF( QUERY( apr_r_i <* apr.representation.items |
         apr_r_i.name = 'latest end' )) > 0
               )) > 0 ) )) = 1;
WR5:   (SIZEOF( QUERY( ap <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
           'ACTION_PROPERTY.DEFINITION') |
         ap.name = 'retention period' )) <= 1 ) AND
       ((SIZEOF( QUERY( ap <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
            'ACTION_PROPERTY.DEFINITION') |
          ap.name = 'retention period' )) = 0 ) OR
        (SIZEOF( QUERY( ap <* USEDIN( SELF,'AUTOMOTIVE_DESIGN.' +
            'ACTION_PROPERTY.DEFINITION') |
        (ap.name = 'retention period') AND
        (SIZEOF( USEDIN( ap, 'AUTOMOTIVE_DESIGN.' +
         'ACTION_PROPERTY_REPRESENTATION.PROPERTY')) = 1 ) AND
        (SIZEOF( QUERY( apr <* USEDIN( ap, 'AUTOMOTIVE_DESIGN.' +
               'ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
        (SIZEOF(apr.representation.items) <= 2) AND
        (SIZEOF( QUERY( apr_r_i <* apr.representation.items |
        (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.TIME_MEASURE_WITH_UNIT']
         * TYPEOF(apr_r_i)) = 2 ) AND
        (apr_r_i.name IN ['earliest end',
          'latest end'])
               )) = SIZEOF(apr.representation.items) ) AND
        (SIZEOF( QUERY( apr_r_i <* apr.representation.items |
         apr_r_i.name = 'earliest end' )) <= 1 ) AND
        (SIZEOF( QUERY( apr_r_i <* apr.representation.items |
         apr_r_i.name = 'latest end' )) <= 1 )
               )) = 1 )
        )) = 1 ));
WR6:   (SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
               'ACTION_RELATIONSHIP.RELATED_ACTION')) = 0 ) AND
       (SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
               'ACTION_RELATIONSHIP.RELATING_ACTION')) <= 3 ) AND
       (SIZEOF( QUERY( ar <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'ACTION_RELATIONSHIP.RELATING_ACTION') |
         (ar.name IN ['start','earliest end','latest end']) AND
         ('AUTOMOTIVE_DESIGN.EVENT_REFERENCE' IN
          TYPEOF(ar.related_action))
              )) = SIZEOF( USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
```

```
          'ACTION_RELATIONSHIP.RELATING_ACTION')) ) AND
        (SIZEOF( QUERY( ar <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'ACTION_RELATIONSHIP.RELATING_ACTION') |
          ar.name = 'start' )) <= 1 ) AND
        (SIZEOF( QUERY( ar <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'ACTION_RELATIONSHIP.RELATING_ACTION') |
          ar.name = 'earliest end' )) <= 1 ) AND
        (SIZEOF( QUERY( ar <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'ACTION_RELATIONSHIP.RELATING_ACTION') |
          ar.name = 'latest end' )) <= 1 );
END_ENTITY;
(*
```

Formal propositions:

WR1: The **retention** shall be referenced as the **assigned_action** by at least one instance of **retention_assignment.**

WR2: The **retention** shall either be contained in the set of **items** of an **applied_date_assignment** that references as its **role** a **date_role** with a **name** of 'start', or shall be contained in the set of **items** of an **applied_date_and_time_assignment** that references as its **role** a **date_time_role** with a **name** of 'start', or shall be referenced as the **relating_action** by an instance of **action_relationship** that has a **name** of 'start'.

WR3: The **retention** shall either be contained in the set of **items** of an **applied_date_assignment** that references as its **role** a **date_role** with a **name** of 'earliest end', or shall be contained in the set of **items** of an **applied_date_and_time_assignment** that references as its **role** a **date_time_role** with a **name** of 'earliest end', or shall be referenced as the **relating_action** by an instance of **action_relationship** that has a **name** of 'earliest end', or shall be referenced as the **definition** by an instance of **action_property** that has a **name** of 'retention period' and that is referenced as the **property** by an instance of **action_property_representation** that references as its **representation** a **representation** that contains a **representation_item** in its set of **items** that has a **name** of 'earliest end'.

WR4: The **retention** shall either be contained in the set of **items** of an **applied_date_assignment** that references as its **role** a **date_role** with a **name** of 'latest end', or shall be contained in the set of **items** of an **applied_date_and_time_assignment** that references as its **role** a **date_time_role** with a **name** of 'latest end', or shall be referenced as the **relating_action** by an instance of **action_relationship** that has a **name** of 'latest end', or shall be referenced as the **definition** by an instance of **action_property** that has a **name** of 'retention period' and that is referenced as the **property** by an instance of **action_property_representation** that references as its **representation** a **representation** that contains a **representation_item** in its set of **items** that has a **name** of 'latest end'.

WR5: There shall be at most one instance of **action_property** with a **name** of 'retention period' that references the **retention** as its **definition**. If present, this **action_property** shall be referenced as the **property** by exactly one instance of **action_property_representation.** This **action_property_representation** shall reference as its **representation** a **representation** that contains at most two **representation_items** in its set of **items**. Both **representa-**

tion_items shall be of type **measure_representation_item** and **time_measure_with_unit** and shall have a **name** of either 'earliest end' or 'latest end'. At most one of the two **representation_item**s shall have a **name** of 'earliest end', and at most one of the two **representation_item**s shall have a **name** of 'latest end'.

WR6: The **retention** shall be referenced by **action_relationship**s only as the **relating_action.** The **retention** shall participate as the **relating_action** in at most three **action_relationship**s. If present, these **action_relationship**s shall have a **name** of either 'start', 'earliest end' or 'latest end' and shall reference as the **related_action** an **event_reference.** The **retention** shall participate in at most one **action_relationship** that has a **name** of 'start', in at most one **action_relationship** that has a **name** of 'earliest end', and in at most one **action_relationship** that has a **name** of 'latest end'.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **retention** entity:

– restrict_multi_language_for_retention (see 5.2.4.157);

– retention_requires_retention_assignment (see 5.2.4.183).

5.2.3.1.124 retention_assignment

A **retention_assignment** is a type of **action_assignment** that assigns a **retention** to a **retention_item** that needs to be retained. See ARM definition for Retention_period in paragraph 4.2.381 for more information.

EXPRESS specification:

```
*)
ENTITY retention_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF retention_item;
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.RETENTION' IN TYPEOF(SELF.assigned_action);
  WR2:  SELF.role.name = 'retention';
END_ENTITY;
(*
```

Attribute definitions:

items: The set of items to be retained.

Formal propositions:

WR1: The **retention_assignment** shall reference as its **assigned_action** an **action** that is of type **retention**.

WR2: The **retention_assignment** shall reference as its **role** an **action_role** that has a **name** of 'retention'.

5.2.3.1.125 rib

A **rib** is a type of **feature_definition** that is the representation of a volume that is added to a solid base shape in order to increase the stability of certain regions of the part. This added volume is represented by sweeping an open profile along a path represented by a **path_feature_component.** The walls of a rib may have an additional slope that is represented by a **taper**. See ARM definition for Rib_feature in paragraph 4.2.382 for more information.

EXPRESS specification:

```
*)
ENTITY rib
  SUBTYPE OF (feature_definition);
WHERE
  WR1:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
         ({1 <= SIZEOF(pdr.used_representation.items) <= 3})
               )) = 1
              )) = 1;
  WR2:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
                NOT (srwp_i.name IN ['orientation','fillet radius'])
               )) > 0 )
               )) > 0
              )) = 0;
  WR3:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
                (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(srwp_i)) = 2 ) AND
                (srwp_i.name = 'fillet radius')
               )) = 1 )
               )) <= 1
```

```
            )) <= 1;
WR4:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
       SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
       TYPEOF(pdr.used_representation)) AND
              (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
              (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(srwp_i)) = 2 ) AND
              (srwp_i.name = 'shoulder radius')
             )) = 1 )
             )) <= 1
            )) <= 1;
WR5:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
           'PROPERTY_DEFINITION.DEFINITION') |
            ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pds)) AND
            (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
       (sa.name = 'rib profile') AND
       (sa.description = 'open profile occurrence') AND
       (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
         TYPEOF(sar)))=1) AND
       (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
          TYPEOF(sar)) AND
        (sar.name = 'open profile usage') AND
        (SIZEOF(['AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE',
         'AUTOMOTIVE_DESIGN.PARTIAL_CIRCULAR_PROFILE',
         'AUTOMOTIVE_DESIGN.ROUNDED_U_PROFILE',
         'AUTOMOTIVE_DESIGN.VEE_PROFILE',
         'AUTOMOTIVE_DESIGN.TEE_PROFILE',
         'AUTOMOTIVE_DESIGN.OPEN_PATH_PROFILE'] *
         TYPEOF(sar.relating_shape_aspect)
              ) = 1 )
              )) = 1 )
             )) = 1 )
            )) = 1;
WR6:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
           'PROPERTY_DEFINITION.DEFINITION') |
            ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pds)) AND
            (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
       (sa.name = 'sweep path') AND
       (sa.description = 'path feature component occurrence') AND
       (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
```

```
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
          'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
           TYPEOF(sar)))=1) AND
         (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
         'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
          ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
            TYPEOF(sar)) AND
          (sar.name = 'path feature component usage') AND
          ('AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN
           TYPEOF(sar.relating_shape_aspect))
                )) = 1 )
               )) = 1 )
              )) = 1;
  WR7:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION.DEFINITION') |
              ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
         TYPEOF(pds)) AND
              (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
               (sa.name = 'taper') AND
               (sa.description = 'taper occurrence') AND
               (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
                'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
         TYPEOF(sar)))=1) AND
               (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
                ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
          TYPEOF(sar)) AND
                (sar.name = 'taper usage') AND
                ('AUTOMOTIVE_DESIGN.TAPER' IN
         TYPEOF(sar.relating_shape_aspect)) AND
                (sar.relating_shape_aspect.description = 'directed taper')
              )) = 1 )
              )) = 1 )
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The implicit representation of the **rib** shall contain between one and three **representation_items** in its set of items.

WR2: The implicit representation of a **rib** shall contain only **representation_items** in its set of **items** that have a **name** of either 'orientation', 'fillet radius', or 'shoulder radius'.

WR3: At most one **representation_item** used for the implicit representation of the **rib** shall have a **name** of 'fillet radius'. If present, this **representation_item** shall be of type **measure_representation_item** and **length_measure_with_unit**.

WR4: At most one **representation_item** used for the implicit representation of the **rib** shall have a **name** of 'shoulder radius'. If present, this **representation_item** shall be of type **measure_representation_item** and **length_measure_with_unit**.

WR5: The **rib** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'rib profile' and a **description** of 'open profile occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'open profile usage' and a **relating_shape_aspect** that references a **square_u_profile, partial_circular_profile, rounded_u_profile, vee_profile, tee_profile,** or **open_path_profile.**

WR6: The **rib** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'sweep path' and a **description** of 'path feature component occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'path feature component usage' and a **relating_shape_aspect** that references a **path_feature_component.**

WR7: The **rib** shall be the basis shape for at most one **shape_aspect** with a **name** of 'taper' and a **description** of 'taper occurrence'. If present, this **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'taper usage' and a **relating_shape_aspect** that references a **taper** with a **description** of 'directed taper'.

Informal propositions:

IP1: The ends of a **rib** shall disappear in the base shape.

5.2.3.1.126 round_hole

A **round_hole** is a type of **feature_definition** that is the representation of a cylindrical enclosed volume that is removed from a solid base shape. This removed volume is represented by sweeping a **circular_closed_profile** along a linear path represented by a **path_feature_component.** The bottom of the **round_hole** is represented by a **hole_bottom.** A **round_hole** may have a change in diameter that is represented by a **taper**. See ARM definition for Round_hole in paragraph 4.2.387 for more information.

EXPRESS specification:

```
*)
ENTITY round_hole
  SUBTYPE OF (feature_definition);
WHERE
  WR1:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION.DEFINITION') |
              ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
         TYPEOF(pds)) AND
              (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
           'SHAPE_ASPECT.OF_SHAPE') |
                (sa.name = 'boundary') AND
                (sa.description = 'closed profile occurrence') AND
```

```
            (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
           'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
             'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) = 1) AND
            (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
           'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
             ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar)) AND
             (sar.name = 'closed profile usage') AND
             ('AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE' IN
      TYPEOF(sar.relating_shape_aspect))
           )) = 1 )
           )) = 1 )
           )) = 1;
WR2:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pds)) AND
           (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
            (sa.name = 'sweep path') AND
            (sa.description = 'path feature component occurrence') AND
            (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
           'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
             'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar)))=1) AND
            (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
           'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
             ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar)) AND
             (sar.name = 'path feature component usage') AND
             ('AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN
      TYPEOF(sar.relating_shape_aspect)) AND
             (sar.relating_shape_aspect.description = 'linear')
           )) = 1 )
           )) = 1 )
           )) = 1;
WR3:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pds)) AND
           (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
          (sa.name = 'bottom condition') AND
          (sa.description = 'hole bottom occurrence') AND
          (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
         'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
           'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(sar)))=1) AND
          (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
           ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
```

```
              TYPEOF(sar)) AND
            (sar.name = 'hole bottom usage') AND
            ('AUTOMOTIVE_DESIGN.HOLE_BOTTOM' IN
              TYPEOF(sar.relating_shape_aspect)) AND
            (sar.relating_shape_aspect.description IN
              ['conical','flat','flat with radius','spherical',
        'through','flat with chamfer'])
           )) = 1 )
            )) = 1 )
             )) = 1;
WR4:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
             ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pds)) AND
             (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
            (sa.name = 'taper') AND
            (sa.description = 'taper occurrence') AND
            (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
           'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
             'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
              TYPEOF(sar))) = 1) AND
            (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
             ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
               TYPEOF(sar)) AND
             (sar.name = 'taper usage') AND
             ('AUTOMOTIVE_DESIGN.TAPER' IN
               TYPEOF(sar.relating_shape_aspect))
           )) = 1 )
            )) <= 1 )
             )) <= 1;
WR5:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
             ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pds)) AND
             (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
            (sa.name = 'top chamfer') AND
            (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
           'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
             'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
              TYPEOF(sar))) = 2) AND
            (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')|
             ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
               TYPEOF(sar)) AND
             (sar.name = 'taper usage') AND
             ('AUTOMOTIVE_DESIGN.CHAMFER_OFFSET' IN
               TYPEOF(sar.related_shape_aspect)) AND
             (sar.related_shape_aspect.name IN
               ['first chamfer offset','second chamfer offset'])
```

```
            )) = 2 )
              )) <= 1 )
              )) <= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **round_hole** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'boundary' and a **description** of 'closed profile occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'closed profile usage' and a **relating_shape_aspect** that references a **circular_closed_profile.**

WR2: The **round_hole** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'sweep path' and a **description** of 'path feature component occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'path feature component usage' and a **relating_shape_aspect** that references a **path_feature_component** with a **description** of 'linear'.

WR3: The **round_hole** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'bottom condition' and a **description** of 'hole bottom occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'hole bottom usage' and the **relating_shape_aspect** that references a **hole_bottom** with a **description** of either 'conical', 'flat', 'flat with radius', 'spherical', 'through', or 'flat with chamfer'.

WR4: The **round_hole** shall be the basis shape for at most one **shape_aspect** with a **name** of 'taper' and a **description** of 'taper occurrence'. If present, this **shape_aspect** shall be the **related_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'taper usage' and the **relating_shape_aspect** that references a **taper**.

WR5: The **round_hole** shall be the basis shape for at most one **shape_aspect** with a **name** of 'top chamfer'. If present, this **shape_aspect** shall be the **relating_shape_aspect** in exactly two **feature_component_relationships** both of which refer to a **chamfer_offset** as **related_shape_aspect.** The first of the two **feature_component_relationships** shall have a **name** of 'first chamfer offset' whereas the second shall have a **name** of 'second chamfer offset'.

5.2.3.1.127 rounded_u_profile

A **rounded_u_profile** is a type of **shape_aspect** that consists of an arc with connected straight lines at each end and a reference system. The reference system consists of two orthogonal reference axes X and Y that intersect in a reference point. See ARM definition for Rounded_U_profile in paragraph 4.2.388 for more information.

EXPRESS specification:

```
*)
ENTITY rounded_u_profile
```

```
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)
         )) = 1;
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
           (SIZEOF(pdr.used_representation.items) = 2)
              )) <= 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
               TYPEOF(pdr.used_representation)) AND
                     (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                             (srwp_i.name = 'orientation') AND
                             ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i))
                            )) = 1)
              )) = 1;
  WR5:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
               TYPEOF(pdr.used_representation)) AND
                     (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                             (srwp_i.name = 'width') AND
              (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
               'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                      * TYPEOF(srwp_i)) = 2 )
                            )) = 1)
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **rounded_u_profile** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **rounded_u_profile** shall have exactly one implicit representation defined by a relationship to a **shape_representation_with_parameters.**

WR3: The **shape_representation_with_parameters** that represents the **rounded_u_profile** shall contain two **representation_items** in its set of **items**.

WR4: Exactly one **representation_item** used for the implicit representation of a **rounded_u_profile** shall be of type **placement** with a **name** of 'orientation'.

WR5: Exactly one **representation_item** used for the implicit representation of a **rounded_u_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'width'.

Informal propositions:

IP1: The **rounded_u_profile** shall be planar and shall lie in the X-Y plane of the reference system.

IP2: The reference point of the **rounded_u_profile** is defined at the midpoint of the arc.

IP3: The **rounded_u_profile** is defined with the X direction tangent to the arc at the midpoint, and the Y direction in the direction of the open end of the profile.

5.2.3.1.128 shape_aspect_transition

A **shape_aspect_transition** is a type of **shape_aspect_relationship** that specifies a relationship between two **shape_aspect**s that represent either two edges or two faces of a product for the purpose of specifying the geometric continuity at the transition of those two **shape_aspects.** See ARM definition for Edge_transition in paragraph 4.2.155 or Face_transition in paragraph 4.2.170 for more information.

EXPRESS specification:

```
*)
ENTITY shape_aspect_transition
  SUBTYPE OF (shape_aspect_relationship);
WHERE
  WR1:  SIZEOF( QUERY( pdr_1 <* USEDIN( SELF.relating_shape_aspect, 'AUTOMOTIVE_DESIGN.' +
                                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           SIZEOF( QUERY( pdr_2 <* USEDIN( SELF.related_shape_aspect,
                                   'AUTOMOTIVE_DESIGN.' +
                                   'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              SIZEOF( QUERY( item_1 <* pdr_1.used_representation.items |
                 SIZEOF( QUERY( item_2 <* pdr_2.used_representation.items |
                         NOT ( NOT ( ('AUTOMOTIVE_DESIGN.EDGE' IN
                       TYPEOF(item_1)) ) XOR
                                     ('AUTOMOTIVE_DESIGN.EDGE' IN
          TYPEOF(item_2)) ) OR
                         NOT ( NOT ( ('AUTOMOTIVE_DESIGN.FACE' IN
          TYPEOF(item_1)) ) XOR
                                     ('AUTOMOTIVE_DESIGN.FACE' IN
                                       TYPEOF(item_2)) ) OR
           ( ('AUTOMOTIVE_DESIGN.FACE' IN
               TYPEOF(item_1)) AND
             ('AUTOMOTIVE_DESIGN.EDGE' IN
               TYPEOF(item_2)) )
                       )) > 0
                  )) > 0
```

```
                )) > 0
              )) = 0;
  WR2:  SIZEOF( QUERY( pdr_1 <* USEDIN( SELF.relating_shape_aspect, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          SIZEOF( QUERY( pdr_2 <* USEDIN( SELF.related_shape_aspect, 'AUTOMOTIVE_DESIGN.'+
                                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
             SIZEOF( QUERY( item_1 <* pdr_1.used_representation.items |
                     ('AUTOMOTIVE_DESIGN.EDGE' IN TYPEOF(item_1) ) AND
                     (SIZEOF( QUERY( item_2 <* pdr_2.used_representation.items |
        ('AUTOMOTIVE_DESIGN.EDGE' IN TYPEOF(item_2)) AND
                                NOT ((item_1\edge.edge_start :=:
        item_2\edge.edge_start) OR
                                   (item_1\edge.edge_start :=:
        item_2\edge.edge_end) OR
                                   (item_1\edge.edge_end :=:
        item_2\edge.edge_start) OR
                                   (item_1\edge.edge_end :=:
        item_2\edge.edge_end)
              )
                             )) > 0)
                   )) > 0
                )) > 0
              )) = 0;
  WR3:  SIZEOF( QUERY( pdr_1 <* USEDIN( SELF.relating_shape_aspect, 'AUTOMOTIVE_DESIGN.' +
                              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          SIZEOF( QUERY( pdr_2 <* USEDIN( SELF.related_shape_aspect,
                                 'AUTOMOTIVE_DESIGN.' +
                                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
             SIZEOF( QUERY( item_1 <* pdr_1.used_representation.items |
                     ('AUTOMOTIVE_DESIGN.FACE' IN TYPEOF(item_1)) AND
                     (SIZEOF( QUERY( item_2 <* pdr_2.used_representation.items |
                              ('AUTOMOTIVE_DESIGN.FACE' IN TYPEOF(item_2)) AND
                              (SIZEOF( QUERY( f_1 <* item_1\face.bounds |
                  ('AUTOMOTIVE_DESIGN.EDGE_LOOP' IN
            TYPEOF(f_1.bound)) AND
                              (SIZEOF( QUERY( f_2 <* item_2\face.bounds |
            ('AUTOMOTIVE_DESIGN.EDGE_LOOP' IN
              TYPEOF(f_2.bound)) AND
            (SIZEOF( QUERY( oe_1 <* f_1.bound\path.edge_list |
                SIZEOF( QUERY( oe_2 <* f_2.bound\path.edge_list |
            oe_1.edge_element :=: oe_2.edge_element )) > 0
                                                 )) > 0)
                                              )) > 0)
                                           )) > 0)
                             )) = 0)
                   )) > 0
                )) > 0
              )) = 0;
  WR4:  SELF.name IN ['g0','g1','g2'];
END_ENTITY;
(*
```

Formal propositions:

WR1: The **shape_aspect**s that play the role of either **relating_shape_aspect** or **related_shape_aspect** shall be associated to **representation**s that contain either an **edge** in both or a **face** in both.

WR2: If the representations contain **edge**s they shall share at least one instance of **vertex**.

WR3: If the representations contain **face**s they shall share at least one instance of **edge**.

WR4: The **shape_aspect_transition** shall have a **name** of either 'g0','g1, or 'g2'.

5.2.3.1.129 shape_defining_relationship

A **shape_defining_relationship** is a type of **shape_aspect_relationship** that is used to relate features with components that define the basic shape of the feature such as profiles or sweep paths.

EXPRESS specification:

```
*)
ENTITY shape_defining_relationship
  SUBTYPE OF (shape_aspect_relationship);
END_ENTITY;
(*
```

5.2.3.1.130 shape_representation_with_parameters

A **shape_representation_with_parameters** is a type of **shape_representation** in which the shape of a **product_definition** or **shape_aspect** is defined implicitly using measurements and descriptive parameters.

EXAMPLE 239 – The shape of a box may be specified by a **shape_representation_with_parameters** by giving measurements for its height, length, and width.

EXPRESS specification:

```
*)
ENTITY shape_representation_with_parameters
  SUBTYPE OF (shape_representation);
WHERE
  WR1:  SIZEOF( QUERY( i <* SELF.items |
                SIZEOF(['AUTOMOTIVE_DESIGN.PLACEMENT',
                        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM']
                       * TYPEOF(i)) = 1
              )) = SIZEOF(SELF.items);
END_ENTITY;
(*
```

Formal propositions:

WR1: The elements that specify the parameters for a **shape_representation_with_parameters** shall be of type placement, **measure_representation_item**, or **descriptive_representation_item.**

5.2.3.1.131 slot

A **slot** is a type of **feature_definition** that is the representation of a volume that is removed from a solid base shape. This removed volume is represented by sweeping an open profile along a path represented by a **path_feature_component.** Each end of the **slot** is represented by a **slot_end.** See ARM definition for Slot_feature in paragraph 4.2.407 for more information.

EXPRESS specification:

```
*)
ENTITY slot
  SUBTYPE OF (feature_definition);
WHERE
  WR1:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION.DEFINITION') |
               SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
               TYPEOF(pdr.used_representation)) AND
              ({1 <= SIZEOF(pdr.used_representation.items) <= 2})
             )) = 1
              )) = 1;
  WR2:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION.DEFINITION') |
               SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
                (pdr.used_representation.name = 'depth') AND
                (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
                (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                * TYPEOF(srwp_i)) = 2 )
               )) = 1 )
               )) <= 1
              )) <= 1;
  WR3:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
             'PROPERTY_DEFINITION.DEFINITION') |
               ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
               (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
             'SHAPE_ASPECT.OF_SHAPE') |
               (sa.name = 'sweep shape') AND
               (sa.description = 'open profile occurrence') AND
```

```
            (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
             'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) = 1 ) AND
            (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
             ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar)) AND
             (sar.name = 'open profile usage') AND
             (SIZEOF(['AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE',
      'AUTOMOTIVE_DESIGN.PARTIAL_CIRCULAR_PROFILE',
      'AUTOMOTIVE_DESIGN.ROUNDED_U_PROFILE',
      'AUTOMOTIVE_DESIGN.VEE_PROFILE',
      'AUTOMOTIVE_DESIGN.TEE_PROFILE',
      'AUTOMOTIVE_DESIGN.OPEN_PATH_PROFILE']
             * TYPEOF(sar.relating_shape_aspect)) = 1 )
           )) = 1 )
            )) = 1 )
           )) = 1;
WR4:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
      ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pds)) AND
      (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
      (sa.name = 'sweep path') AND
      (sa.description = 'path feature component occurrence') AND
      (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
             'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
       'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
        TYPEOF(sar))) = 1 ) AND
      (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
       ('AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN
         TYPEOF(sar)) AND
       (sar.name = 'path feature component usage') AND
       ('AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN
        TYPEOF(sar.relating_shape_aspect))
             )) = 1 )
            )) = 1 )
           )) = 1;
WR5:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
          'PROPERTY_DEFINITION.DEFINITION') |
      ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pds)) AND
      (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') |
      (sa.name = 'end condition at path start') AND
      (sa.description = 'slot end occurrence') AND
      (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
             'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
       'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
```

```
          TYPEOF(sar))) = 1 ) AND
        (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
         ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
           TYPEOF(sar)) AND
         (sar.name = 'slot end usage') AND
         ('AUTOMOTIVE_DESIGN.SLOT_END' IN
          TYPEOF(sar.relating_shape_aspect))
               )) = 1 )
              )) = 1 )
             )) = 1;
  WR6:  SIZEOF( QUERY( pds <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pds)) AND
        (SIZEOF( QUERY( sa <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
          'SHAPE_ASPECT.OF_SHAPE') |
        (sa.name = 'end condition at path end') AND
        (sa.description = 'slot end occurrence') AND
        (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
               'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
         'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
          TYPEOF(sar))) = 1 ) AND
        (SIZEOF( QUERY( sar <* USEDIN( sa, 'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
         ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
           TYPEOF(sar)) AND
         (sar.name = 'slot end usage') AND
         ('AUTOMOTIVE_DESIGN.SLOT_END' IN
          TYPEOF(sar.relating_shape_aspect))
               )) = 1 )
              )) = 1 )
             )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The implicit representation of the **slot** shall contain one or two **representation_items** in its set of **items**.

WR2: At most one **representation_item** used for the implicit representation of the **slot** shall have a **name** of 'depth'. If present, this **representation_item** shall be of type **measure_representation_item** and **length_measure_with_unit** with.

WR3: The **slot** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'sweep shape' and a **description** of 'open profile occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'open profile usage' and a **relating_shape_aspect** that references a **square_u_profile**, **partial_circular_profile**, **rounded_u_profile**, **vee_profile**, **tee_profile**, or **open_path_profile.**

WR4: The **slot** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'sweep path' and a **description** of 'path feature component occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **shape_defining_relationship** with a **name** of 'path feature component usage' and a **relating_shape_aspect** that references a **path_feature_component.**

WR5: The **slot** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'end condition at path start' and a **description** of 'slot end occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'slot end usage' in which the **relating_shape_aspect** is a **slot_end.**

WR6: The **slot** shall be the basis shape for exactly one **shape_aspect** with a **name** of 'end condition at path end' and a **description** of 'slot end occurrence'. This **shape_aspect** shall be the **related_shape_aspect** in exactly one **feature_component_relationship** with a **name** of 'slot end usage' in which the **relating_shape_aspect** is a **slot_end.**

5.2.3.1.132 slot_end

A **slot_end** is a type of **shape_aspect** that is the representation of the end condition for a **slot**. For more information see ARM definition for Slot_end_type in paragraph 4.2.406, for Flat_slot_end_type in paragraph 4.2.184, for Open_slot_end_type in paragraph 4.2.293, for Radiused_slot_end_type in paragraph 4.2.367, and for Woodruff_slot_end_type in paragraph 4.2.499.

EXPRESS specification:

```
*)
ENTITY slot_end
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SELF.description IN ['open','radiused','flat','woodruff'];
  WR3:  NOT (SELF.description IN ['open','radiused']) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation) )) = 0 );
  WR4:  NOT (SELF.description IN ['flat','woodruff']) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
             TYPEOF(pdr.used_representation)
               )) = 1 );
  WR5:  (SELF.description <> 'flat') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
          (SIZEOF(pdr.used_representation.items) = 2)
               )) = 1 );
```

```
WR6:  (SELF.description <> 'flat') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
         (srwp_i.name = 'first radius') AND
         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
          'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
         * TYPEOF(srwp_i)) = 2 )
         )) = 1)
             )) = 1 );
WR7:  (SELF.description <> 'flat') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
         (srwp_i.name = 'second radius') AND
         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
          'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
         * TYPEOF(srwp_i)) = 2 )
         )) = 1)
             )) = 1 );
WR8:  (SELF.description <> 'woodruff') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        (SIZEOF(pdr.used_representation.items) = 1)
             )) = 1 );
WR9:  (SELF.description <> ' woodruff ') OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY(srwp_i <* pdr.used_representation.items |
         (srwp_i.name = 'radius') AND
         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
          'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
         * TYPEOF(srwp_i)) = 2 )
         )) = 1)
             )) = 1 );
WR10: SIZEOF( QUERY( fcr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
         ('AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN
           TYPEOF(fcr)) AND
         (fcr.description = 'slot end usage') AND
         ('AUTOMOTIVE_DESIGN.SLOT' IN TYPEOF(fcr.related_shape_aspect)) AND
       (fcr.related_shape_aspect.name IN ['end condition at path start',
          'end condition at path end']) AND
       (fcr.related_shape_aspect.description = 'slot end occurrence')
```

```
              )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **slot_end** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **description** for the **slot_end** shall be either 'open', 'radiused', 'flat', or 'woodruff'.

WR3: If the **description** of the **slot_end** is 'open' or 'radiused', the **slot_end** shall not have any implicit representation.

WR4: If the **description** of the **slot_end** is 'flat' or 'woodruff' the **slot_end** shall have its implicit representation specified by exactly one **shape_representation_with_parameters.**

WR5: If the **description** of the **slot_end** is 'flat', the **slot_end** shall have an implicit representation that contains exactly two **representation_items** in its set of **items**.

WR6: If the **description** of the **slot_end** is 'flat', exactly one **representation_item** used for the implicit representation of the **slot_end** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'first radius'.

WR7: If the **description** of the **slot_end** is 'flat', exactly one **representation_item** used for the implicit representation of the **slot_end** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'second radius'.

WR8: If the **description** of the **slot_end** is 'woodruff', the **slot_end** shall have an implicit representation that contains exactly one **representation_item** in its set of **items**.

WR9: If the **description** of the **slot_end** is 'woodruff', exactly one **representation_item** used for the implicit representation of the **slot_end** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'radius'.

WR10: The **slot_end** shall be the **relating_shape_aspect** in exactly one **feature_component_relationship** with a **description** of 'slot end usage' in which the **related_shape_aspect** is an aspect of the shape of a **slot** with a **name** of either 'end condition at path start' or 'end condition at path end' and a **description** of 'slot end occurrence'.

5.2.3.1.133 solution_alternative_definition

A **solution_alternative_definition** is a type of **configuration_design** that identifies a **product_component** to which a **solution_based_product_definition** provides a solution. The **solution_alternative_definition**s associated to the same **product_component** represent mutually exclusive alternatives.

EXPRESS specification:

```
*)
ENTITY solution_alternative_definition
  SUBTYPE OF (configuration_design);
```

```
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.SOLUTION_BASED_PRODUCT_DEFINITION' IN
        TYPEOF(SELF.design);
  WR2:  'AUTOMOTIVE_DESIGN.PRODUCT_COMPONENT' IN
        TYPEOF(SELF.configuration);
END_ENTITY;
(*
```

Formal propositions:

WR1: The **solution_alternative_definition** shall reference as the **design** a **product_definition** that is of type **solution_based_product_definition.**

WR2: The **solution_alternative_definition** shall reference as the **configuration** a **configuration_item** that is of type **product_component.**

5.2.3.1.134 solution_based_product_definition

A **solution_based_product_definition** is a type of **product_definition** that identifies a solution to a functional requirement. It specifies a set of **product_definition_occurrence** instances that jointly fulfill this functional requirement. See ARM definition for Item_solution in paragraph 4.2.246, for Technical_solution in paragraph 4.2.451, for Supplier_solution in paragraph 4.2.433, and for Final_item_solution in paragraph 4.2.182 for more information.

EXPRESS specification:

```
*)
ENTITY solution_based_product_definition
  SUBTYPE OF (product_definition);
WHERE
  WR1:  (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                 'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
                 (pdr.name = 'solution alternative definition') AND
                 ('AUTOMOTIVE_DESIGN.SOLUTION_BASED_PRODUCT_DEFINITION' IN
                   TYPEOF(pdr.relating_product_definition)) )) +
         SIZEOF( QUERY( cd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                      'CONFIGURATION_DESIGN.DESIGN') |
                 'AUTOMOTIVE_DESIGN.SOLUTION_ALTERNATIVE_DEFINITION' IN
                  TYPEOF(cd)))) = 1;
  WR2:  SELF.name IN ['final','supplier','technical','technical supplier',
                      'technical final','supplier final','technical supplier final'];
  WR3:  NOT (SELF.name IN ['supplier','technical supplier','supplier final',
                           'technical supplier final']) OR
        (SIZEOF( QUERY( aoa <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                       'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
                 aoa.role.name = 'supplier'
               )) = 1 );
  WR4:  NOT (SELF.name IN ['final','technical final','supplier final',
                           'technical supplier final']) OR
```

```
        (SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                      'PROPERTY_DEFINITION.DEFINITION') |
                 SIZEOF( QUERY( pdr <* USEDIN(pd, 'AUTOMOTIVE_DESIGN.' +
                                       'PROPERTY_DEFINITION_REPRESENTATION.' +
                                       'DEFINITION') |
                         (pdr.used_representation.description =
                          'final item characteristics') AND
                         (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                                  ('AUTOMOTIVE_DESIGN.' +
                                   'DESCRIPTIVE_REPRESENTATION_ITEM' IN
                                   TYPEOF(r_i)) AND
                                  (r_i.name = 'final item status')
                                )) = 1 )
                       )) = 1
                 )) = 1 );
  WR5:  NOT (SELF.name IN ['final','technical final','supplier final',
                           'technical supplier final']) OR
        (SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                     'PROPERTY_DEFINITION.DEFINITION') |
                 SIZEOF( QUERY( pdr <* USEDIN(pd, 'AUTOMOTIVE_DESIGN.' +
                                       'PRODUCT_DEFINITION_RELATIONSHIP.' +
                                       'RELATING_PRODUCT_DEFINITION') |
                         (pdr.name = 'final specification') AND
                         (pdr.related_product_definition.frame_of_reference.name =
                          'definitional template')
                       )) = 1
                 )) >= 1 ) OR
        (SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                      'PROPERTY_DEFINITION.DEFINITION') |
                 SIZEOF( QUERY( pdr <* USEDIN(pd, 'AUTOMOTIVE_DESIGN.' +
                                       'PROPERTY_DEFINITION_REPRESENTATION.' +
                                       'DEFINITION') |
                         (pdr.used_representation.description =
                          'final item characteristics') AND
                         (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                                  ('AUTOMOTIVE_DESIGN.' +
                                   'DESCRIPTIVE_REPRESENTATION_ITEM' IN
                                   TYPEOF(r_i)) AND
                                  (r_i.name = 'final specification')
                                )) >= 1 )
                       )) = 1
                 )) > 0 );
END_ENTITY;
(*
```

Formal propositions:

WR1: The **solution_based_product_definition** shall either be referenced as the **related_product_definition** by exactly one **product_definition_relationship** that has a **name** of 'solution alternative definition' and that references as the **relating_product_definition** a **solution_based_product_definition**, or shall be referenced as the **design** by exactly one **confi-**

guration_design that is of type **solution_alternative_definition.**

WR2: The **solution_based_product_definition** shall have a **name** of either 'final', 'supplier', 'technical', 'technical supplier', 'technical final', 'supplier final', or 'technical supplier final'.

WR3: If the **solution_based_product_definition** has a **name** of 'supplier', 'technical supplier', 'supplier final', or 'technical supplier final' it shall be contained in the set of **items** of exactly one **applied_organization_assignment** that references as its **role** an **organization_role** with a **name** of 'supplier'.

WR4: If the **solution_based_product_definition** has a **name** of 'final', 'technical final', 'supplier final', or 'technical supplier final' it shall be referenced as the **definition** by a **property_definition** that is referenced as the **definition** by a **property_definition_representation** that references as its **used_representation** a **representation** that has a **description** of 'final item characteristics'. This **representation** shall contain one **descriptive_representation_item** in its set of **items** that has a **name** of 'final item status'.

WR5: If the **solution_based_product_definition** has a **name** of 'final', 'technical final', 'supplier final', or 'technical supplier final', it shall be referenced as the **relating_product_definition** by at least one **product_definition_relationship** with a **name** of 'final specification' that references as the **related_product_definition** a **product_definition** that references as its **frame_of_reference** a **product_definition_context** with a **name** of 'definitional template', or it shall be referenced as the **definition** by a **property_definition** that is referenced as the **definition** by a **property_definition_representation** that references as its **used_representation** a **representation** that has a **description** of 'final item characteristics' and contains at least one **descriptive_representation_item** in its set of **items** that has a **name** of 'final specification'.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **solution_based_product_definition** entity:

– restrict_multi_language_for_solution_based_product_definition (see 5.2.4.163);

– restrict_version_assignment_for_solution_based_product_definition (see 5.2.4.182).

5.2.3.1.135 square_u_profile

A **square_u_profile** is a type of **shape_aspect** that consists of three connected straight lines and a reference system for the parameters of the **square_u_profile.** The reference system is used to define location and orientation of the **square_u_profile** and consists of two orthogonal reference axes X and Y that intersect in a reference point. See ARM definition for Square_u_profile in paragraph 4.2.423 for more information.

EXPRESS specification:

```
*)
ENTITY square_u_profile
  SUBTYPE OF (shape_aspect);
```

```
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)
         )) = 1;
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
           ({4 <= SIZEOF(pdr.used_representation.items) <= 6})
              )) = 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         NOT (srwp_i.name IN ['orientation','width',
                                              'first angle','second angle',
                                              'first radius','second radius'])
                       )) > 0)
              )) = 0;
  WR5:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'orientation') AND
                         ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i))
                       )) = 1)
              )) = 1;
  WR6:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'width') AND
              (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
           'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
            * TYPEOF(srwp_i)) = 2 )
                       )) = 1)
              )) = 1;
  WR7:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'first angle') AND
              (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
           'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
```

```
          * TYPEOF(srwp_i)) = 2 )
                      )) = 1)
             )) = 1;
  WR8:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
               (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                        (srwp_i.name = 'second angle') AND
             (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
          'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
          * TYPEOF(srwp_i)) = 2 )
                      )) = 1)
             )) = 1;
  WR9:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
               (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                        (srwp_i.name = 'first radius') AND
             (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
          'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
          * TYPEOF(srwp_i)) = 2 )
                      )) = 1)
             )) <= 1;
  WR10: SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
               (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                        (srwp_i.name = 'second radius') AND
             (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
          'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
          * TYPEOF(srwp_i)) = 2 )
                      )) = 1)
             )) <= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **square_u_profile** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **square_u_profile** shall have exactly one implicit representation defined by a **shape_representation_with_parameters.**

WR3: The **shape_representation_with_parameters** that represents the **square_u_profile** shall contain between four and six **representation_items** in its set of **items**.

WR4: The implicit representation of a **square_u_profile** shall contain only **representation_items** in its set of **items** that have a **name** of either 'orientation', 'width', 'first angle', 'second

angle', 'first radius', or 'second radius'.

WR5: Exactly one **representation_item** used for the implicit representation of a **square_u_profile** shall be of type **placement** with a **name** of 'orientation'.

WR6: Exactly one **representation_item** used for the implicit representation of a **square_u_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'width'.

WR7: Exactly one **representation_item** used for the implicit representation of a **square_u_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'first angle'.

WR8: Exactly one **representation_item** used for the implicit representation of a **square_u_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'second angle'.

WR9: At most one **representation_item** used for the implicit representation of a **square_u_profile** shall have a **name** of 'first radius'. If present, this **representation_item** shall be of type **measure_representation_item** and **length_measure_with_unit.**

WR10: At most one **representation_item** used for the implicit representation of a **square_u_profile** shall have a **name** of 'second radius'. If present, this **representation_item** shall be of type **measure_representation_item** and **length_measure_with_unit.**

Informal propositions:

IP1: The **square_u_profile** shall be planar and shall lie in the X-Y plane of the reference system.

IP2: The reference point of the **square_u_profile** is defined at the midpoint of the base line.

IP3: The **square_u_profile** is defined with the base line in the X direction and the Y direction in the direction of the open end of the profile.

5.2.3.1.136 surface_condition

A **surface_condition** is a type of **property_definition** that has a **representation** specifying surface condition information. See ARM definition for Surface_condition in paragraph 4.2.437, for Coating_layer in paragraph 4.2.74, for Contact_ratio in paragraph 4.2.97, for Hardness in paragraph 4.2.225, for Roughness in paragraph 4.2.386, for Tactile_appearance in paragraph 4.2.445, for Treatment_result in paragraph 4.2.477, and for Visual_appearance in 4.2.496 for more information.

EXPRESS specification:

```
*)
ENTITY surface_condition
  SUBTYPE OF (property_definition);
WHERE
```

```
WR1:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               NOT (pdr.used_representation.description IN
                    ['coating layer','contact ration','hardness','roughness',
                     'tactile appearance','treatment result','visual appearance'])
            )) = 0;
WR2:  SIZEOF(['AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE',
               'AUTOMOTIVE_DESIGN.SHAPE_ASPECT']
             * TYPEOF(SELF.definition)) = 1;
WR3:  (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               pdr.used_representation.desciption = 'coating layer' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               (pdr.used_representation.desciption = 'coating layer') AND
               (SIZEOF(pdr.used_representation.items) = 1) AND
               (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                        (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                 'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                * TYPEOF(r_i)) = 2 ) AND
                        (r_i.name = 'thickness')
                      )) = 1 )
             )) = SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                               pdr.used_representation.desciption = 'coating layer' )));
WR4:  (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               pdr.used_representation.desciption = 'contact ratio' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               (pdr.used_representation.desciption = 'contact ratio') AND
               (SIZEOF( QUERY( pdr1 <* USEDIN(pdr.used_representation,
                                    'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION_' +
                                    'REPRESENTATION.USED_REPRESENTATION') |
                        (pdr1.name = 'measuring reference') AND
                 ('AUTOMOTIVE_DESIGN.SURFACE_CONDITION' IN
                  TYPEOF(pdr1.definition)) AND
                 ('AUTOMOTIVE_DESIGN.PHYSICALLY_MODELLED_PRODUCT_DEFINITION' IN
                  TYPEOF(pdr1.definition\property_definition.definition))
                      )) = 1 )
             )) > 0 );
WR5:  (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               pdr.used_representation.desciption = 'contact ratio' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               (pdr.used_representation.desciption = 'contact ratio') AND
               (SIZEOF(pdr.used_representation.items) = 1) AND
               (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                        (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                 'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                * TYPEOF(r_i)) = 2 ) AND
```

```
                        (r_i.name = 'ratio value')
                      )) = 1 )
              )) = SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                            pdr.used_representation.desciption = 'contact ratio' )));
WR6:  (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'hardness' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'hardness') AND
              ('AUTOMOTIVE_DESIGN.MATERIAL_PROPERTY_REPRESENTATION' IN
               TYPEOF(pdr))
              )) = SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                            pdr.used_representation.desciption = 'hardness' )));
WR7:  (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'hardness' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'hardness') AND
              ({2 <= SIZEOF(pdr.used_representation.items) <= 5}) AND
              (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                        (('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                          TYPEOF(r_i)) AND
                         (r_i.name IN ['description','measuring method',
                                       'measuring position'])) OR
                        (('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN
                          TYPEOF(r_i)) AND
                         (r_i.name = 'hardness value')) OR
                        ((SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                  'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                  * TYPEOF(r_i)) = 2) AND
                         (r_i.name = 'depth'))
                      )) = SIZEOF(pdr.used_representation.items))
              )) = SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                            pdr.used_representation.desciption = 'hardness' )));
WR8:  (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'hardness' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'hardness') AND
              (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                        ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                         TYPEOF(r_i)) AND
                        (r_i.name = 'description') )) = 1 )
              )) <= 1 );
WR9:  (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```

```
                pdr.used_representation.desciption = 'hardness' )) = 0 ) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                (pdr.used_representation.desciption = 'hardness') AND
                (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                         ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                          TYPEOF(r_i)) AND
                         (r_i.name = 'measuring method') )) = 1 )
              )) = 1 );
WR10: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                pdr.used_representation.desciption = 'hardness' )) = 0 ) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                (pdr.used_representation.desciption = 'hardness') AND
                (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                         ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                          TYPEOF(r_i)) AND
                         (r_i.name = 'measuring position') )) = 1 )
              )) <= 1 );
WR11: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                pdr.used_representation.desciption = 'hardness' )) = 0 ) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                (pdr.used_representation.desciption = 'hardness') AND
                (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                  'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                  * TYPEOF(r_i)) = 2) AND
                         (r_i.name = 'measuring position') )) = 1 )
              )) <= 1 );
WR12: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                pdr.used_representation.desciption = 'hardness' )) = 0 ) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                (pdr.used_representation.desciption = 'hardness') AND
                (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                         ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN
                          TYPEOF(r_i)) AND
                         (r_i.name = 'hardness value') )) = 1 )
              )) = 1 );
WR13: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                pdr.used_representation.desciption = 'roughness' )) = 0 ) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                (pdr.used_representation.desciption = 'roughness') AND
                ({2 <= SIZEOF(pdr.used_representation.items) <= 6}) AND
                (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                         (('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
```

```
                          TYPEOF(r_i)) AND
                         (r_i.name = 'measuring method')) OR
                        (('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN
                          TYPEOF(r_i)) AND
                         (r_i.name IN ['roughness max value',
                                       'roughness min value'])) OR
                        ((SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                  'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                  * TYPEOF(r_i)) = 2) AND
                         (r_i.name IN ['cutoff value',
                                       'sampling length'])) OR
                        ((SIZEOF(['AUTOMOTIVE_DESIGN.PCURVE',
                                  'AUTOMOTIVE_DESIGN.SURFACE_CURVE']
                                  * TYPEOF(r_i)) = 1) AND
                         (r_i.name = 'measuring direction'))
                     )) = SIZEOF(pdr.used_representation.items))
            )) = SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                          pdr.used_representation.desciption = 'roughness' )));
WR14: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'roughness' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'roughness') AND
              (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                        ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN
                         TYPEOF(r_i)) AND
                        (r_i.name = 'roughness max value') )) = 1 )
            )) = 1 );
WR15: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'roughness' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'roughness') AND
              (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                        ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN
                         TYPEOF(r_i)) AND
                        (r_i.name = 'roughness min value') )) = 1 )
            )) <= 1 );
WR16: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'roughness' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'roughness') AND
              (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                        (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                 'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                 * TYPEOF(r_i)) = 2) AND
                        (r_i.name = 'cutoff value') )) = 1 )
```

```
            )) <= 1 );
WR17: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'roughness' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'roughness') AND
              (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                       (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                * TYPEOF(r_i)) = 2) AND
                       (r_i.name = 'sampling length') )) = 1 )
            )) <= 1 );
WR18: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'roughness' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'roughness') AND
              (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                       ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                        TYPEOF(r_i)) AND
                       (r_i.name = 'measuring method') )) = 1 )
            )) = 1 );
WR19: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'roughness' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'roughness') AND
              (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                       (SIZEOF(['AUTOMOTIVE_DESIGN.PCURVE',
                                'AUTOMOTIVE_DESIGN.SURFACE_CURVE']
                                * TYPEOF(r_i)) = 1) AND
                       (r_i.name = 'measuring method') )) = 1 )
            )) = 1 );
WR20: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'tactile appearance' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'tactile appearance') AND
              ({1 <= SIZEOF(pdr.used_representation.items) <= 2}) AND
              (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                       (('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                         TYPEOF(r_i)) AND
                        (r_i.name = 'description')) OR
                       ((SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                                 'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                 * TYPEOF(r_i)) = 2) AND
                        (r_i.name = 'depth'))
                     )) = SIZEOF(pdr.used_representation.items))
```

```
            )) = SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                              pdr.used_representation.desciption = 'tactile appearance')));
WR21: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'tactile appearance' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'tactile appearance') AND
              (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                       ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                        TYPEOF(r_i)) AND
                       (r_i.name = 'description') )) = 1 )
            )) <= 1 );
WR22: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'tactile appearance' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'tactile appearance') AND
              (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                       (SIZEOF(['AUTOMOTIVE_DESIGN.PCURVE',
                                'AUTOMOTIVE_DESIGN.SURFACE_CURVE']
                                * TYPEOF(r_i)) = 1) AND
                       (r_i.name = 'depth') )) = 1 )
            )) <= 1 );
WR23: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'treatmen result' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'treatmen result') AND
              ({1 <= SIZEOF(pdr.used_representation.items) <= 2}) AND
              (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                       (('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                         TYPEOF(r_i)) AND
                        (r_i.name IN ['description','purpose']))
                     )) = SIZEOF(pdr.used_representation.items))
            )) = SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                              pdr.used_representation.desciption = 'treatmen result' )));
WR24: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              pdr.used_representation.desciption = 'treatmen result' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (pdr.used_representation.desciption = 'treatmen result') AND
              (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                       ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                        TYPEOF(r_i)) AND
                       (r_i.name = 'description') )) = 1 )
            )) = 1 );
```

```
WR25: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               pdr.used_representation.desciption = 'treatmen result' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               (pdr.used_representation.desciption = 'treatmen result') AND
               (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                        ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                         TYPEOF(r_i)) AND
                        (r_i.name = 'purpose') )) = 1 )
             )) <= 1 );
WR26: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               pdr.used_representation.desciption = 'visual appearance' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               (pdr.used_representation.desciption = 'visual appearance') AND
               ({2 <= SIZEOF(pdr.used_representation.items) <= 5}) AND
               (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                        (('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                          TYPEOF(r_i)) AND
                         (r_i.name IN ['colour id','colour name','lustre',
                                       'pattern','transparency']))
                      )) = SIZEOF(pdr.used_representation.items))
             )) = SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                           pdr.used_representation.desciption = 'visual appearance' )));
WR27: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               pdr.used_representation.desciption = 'visual appearance' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               (pdr.used_representation.desciption = 'visual appearance') AND
               (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                        ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                         TYPEOF(r_i)) AND
                        (r_i.name = 'colour id') )) = 1 )
             )) = 1 );
WR28: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               pdr.used_representation.desciption = 'visual appearance' )) = 0 ) OR
      (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               (pdr.used_representation.desciption = 'visual appearance') AND
               (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                        ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                         TYPEOF(r_i)) AND
                        (r_i.name = 'colour name') )) = 1 )
             )) <= 1 );
WR29: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
               pdr.used_representation.desciption = 'visual appearance' )) = 0 ) OR
```

```
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                  (pdr.used_representation.desciption = 'visual appearance') AND
                  (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                           ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                            TYPEOF(r_i)) AND
                           (r_i.name = 'lustre') )) = 1 )
               )) = 1 );
  WR30: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                  pdr.used_representation.desciption = 'visual appearance' )) = 0 ) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                  (pdr.used_representation.desciption = 'visual appearance') AND
                  (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                           ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                            TYPEOF(r_i)) AND
                           (r_i.name = 'pattern') )) = 1 )
               )) <= 1 );
  WR31: (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                  pdr.used_representation.desciption = 'visual appearance' )) = 0 ) OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                  (pdr.used_representation.desciption = 'visual appearance') AND
                  (SIZEOF( QUERY( r_i <* pdr.used_representation.items |
                           ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                            TYPEOF(r_i)) AND
                           (r_i.name = 'transparency') )) = 1 )
               )) <= 1 );
END_ENTITY;
(*
```

Formal propositions:

WR1: The **surface_condition** shall only be referenced as the **definition** by instances of **property_definition_representation** that reference as their **used_representation** a **representation** that has a **description** of either 'coating layer', 'contact ratio', 'hardness', 'roughness', 'tactile appearance', 'treatment result', or 'visual appearance'.

WR2: The **surface_condition** shall reference as its **definition** a **product_definition_shape** or a **shape_aspect.**

WR3: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'coating layer', this **representation** shall contain exactly one **representation_item** in its set of **items**. This **representation_item** shall be a **measure_representation_item** and **length_measure_with_unit** and shall have a **name** of 'thickness'.

WR4: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a

description of 'contact ratio', this **representation** shall also be referenced as the **used_representation** by exactly one **property_definition_representation** that has a **name** of 'measuring reference'. This **property_definition_representation** shall reference as its **definition** a **surface_condition** that references as its **definition** a **product_definition** that is of type **physically_modelled_product_definition.**

WR5: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'contact ratio', this **representation** shall contain exactly one **representation_item** in its set of **items**. This **representation_item** shall be a **measure_representation_item** and **ratio_measure_with_unit** and shall have a **name** of 'ratio value'.

WR6: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'hardness', this **property_definition_representation** shall be a **material_property_representation.**

WR7: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'hardness', this **representation** shall contain between two and five **representation_items** in its set of **items**. These **representation_items** shall either be of type **descriptive_representation_item** and have a **name** of 'description', 'measuring method', or 'measuring position', or shall be of type **measure_representation_item** and have a **name** of 'hardness value', or shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'depth'.

WR8: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'hardness', this **representation** shall contain at most one **representation_item** in its set of **items** that is of type **descriptive_representation_item** and has a **name** of 'description'.

WR9: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'hardness', this **representation** shall contain exactly one **representation_item** in its set of **items** that is of type **descriptive_representation_item** and has a **name** of 'measuring method'.

WR10: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'hardness', this **representation** shall contain at most one **representation_item** in its set of **items** that is of type **descriptive_representation_item** and has a **name** of 'measuring position'.

WR11: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'hardness', this **representation** shall contain at most one **representation_item** in its set of **items** that is of type **measure_representation_item** and **length_measure_with_unit** and has a **name** of 'depth'.

WR12: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'hardness', this **representation** shall contain exactly one **representation_item** in its set of **items** that is of type **measure_representation_item** and has a **name** of 'hardness value'.

WR13: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'roughness', this **representation** shall contain between two and six **representation_item**s in its set of **items**. These **representation_item**s shall either be of type **descriptive_representation_item** and have a **name** of 'measuring method', or shall be of type **measure_representation_item** and have a **name** of 'roughness max value' or 'roughness min value', or shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'cutoff value' or 'sampling length', or shall be of type **pcurve** or **surface_curve** with a **name** of 'measuring direction'.

WR14: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'roughness', this **representation** shall contain exactly one **representation_item** in its set of **items** that is of type **measure_representation_item** and has a **name** of 'roughness max value'.

WR15: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'roughness', this **representation** shall contain at most one **representation_item** in its set of **items** that is of type **measure_representation_item** and has a **name** of 'roughness min value'.

WR16: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'roughness', this **representation** shall contain at most one **representation_item** in its set of **items** that is of type **measure_representation_item** and **length_measure_with_unit** and has a **name** of 'cutoff value'.

WR17: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'roughness', this **representation** shall contain at most one **representation_item** in its set of **items** that is of type **measure_representation_item** and **length_measure_with_unit** and has a **name** of 'sampling length'.

WR18: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'roughness', this **representation** shall contain exactly one **representation_item** in its set of **items** that is of type **descriptive_representation_item** and has a **name** of 'measuring method'.

WR19: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'roughness', this **representation** shall contain exactly one **representation_item** in its set of **items** that is of type **pcurve** or **surface_curve** and has a **name** of 'measuring

method'.

WR20: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'tactile appearance', this **representation** shall contain between one and two **representation_item**s in its set of **items**. These **representation_item**s shall either be of type **descriptive_representation_item** and have a **name** of 'description', or shall be of type **measure_representation_item** and **length_measure_with_unit** and have a **name** of 'depth'.

WR21: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'tactile appearance', this **representation** shall contain at most one **representation_item** in its set of **items** that is of type **descriptive_representation_item** and has a **name** of 'description'.

WR22: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'tactile appearance', this **representation** shall contain at most one **representation_item** in its set of **items** that is of type **measure_representation_item** and **length_measure_with_unit** and has a **name** of 'depth'.

WR23: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'treatment result', this **representation** shall contain between one and two **representation_item**s in its set of **items**. These **representation_item**s shall be of type **descriptive_representation_item** and have a **name** of 'description' or 'purpose'.

WR24: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'treatment result', this **representation** shall contain exactly one **representation_item** in its set of **items** that is of type **descriptive_representation_item** and has a **name** of 'description'.

WR25: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'treatment result', this **representation** shall contain at most one **representation_item** in its set of **items** that is of type **descriptive_representation_item** and has a **name** of 'purpose'.

WR26: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'visual appearance', this **representation** shall contain between two and five **representation_item**s in its set of **items**. These **representation_item**s shall be of type **descriptive_representation_item** and have a **name** of 'colour id', 'colour name', 'lustre', 'pattern', or 'transparency'.

WR27: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'visual appearance', this **representation** shall contain exactly one **representation_item** in its set of **items** that is of type **descriptive_representation_item** and has a

name of 'colour id'.

WR28: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'visual appearance', this **representation** shall contain at most one **representation_item** in its set of **items** that is of type **descriptive_representation_item** and has a **name** of 'colour name'.

WR29: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'visual appearance', this **representation** shall contain exactly one **representation_item** in its set of **items** that is of type **descriptive_representation_item** and has a **name** of 'lustre'.

WR30: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'visual appearance', this **representation** shall contain at most one **representation_item** in its set of **items** that is of type **descriptive_representation_item** and has a **name** of 'pattern'.

WR31: If the **surface_condition** is referenced as the **definition** by a **property_definition_representation** that references as the **used_representation** a **representation** that has a **description** of 'visual appearance', this **representation** shall contain at most one **representation_item** in its set of **items** that is of type **descriptive_representation_item** and has a **name** of 'transparency'.

5.2.3.1.137 surface_condition_callout

A **surface_condition_callout** is a type of **draughting_callout** that presents a surface condition.

EXPRESS specification:

```
*)
ENTITY surface_condition_callout
  SUBTYPE OF (draughting_callout);
END_ENTITY;
(*
```

5.2.3.1.138 taper

A **taper** is a type of **shape_aspect** which represents a linear change of shape applied to a feature. See ARM definition for Angle_taper in paragraph 4.2.10, for Diameter_taper in paragraph 4.2.138, and for Directed_taper in paragraph 4.2.145 for more information.

EXPRESS specification:

```
*)
```

```
ENTITY taper
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SELF.description IN ['angle taper','diameter taper','directed taper'];
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)
              )) = 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
         (SIZEOF(pdr.used_representation.items) = 1)
              )) = 1;
  WR5:  (SELF.description <> 'angle taper') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
           (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
            'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 ) AND
           (srwp_i.name = 'taper angle')
         )) = 1 )
               )) = 1 );
  WR6:  (SELF.description <> 'diameter taper') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
            'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 ) AND
           (srwp_i.name = 'final diameter')
         )) = 1 )
               )) = 1 );
  WR7:  (SELF.description <> 'directed taper') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
          (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
           (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
           'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
          * TYPEOF(srwp_i)) = 2 ) AND
           (srwp_i.name = 'draft angle')
         )) = 1 )
```

```
              )) = 1 );
  WR8:  (SELF.description <> 'directed taper') OR
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN
            TYPEOF(pdr.used_representation))
              )) = 1 ) AND
        (SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN
            TYPEOF(pdr.used_representation)) AND
          (pdr.used_representation.description = 'draft direction')
              )) = 1 );
END_ENTITY;
(*
```

Formal propositions:

WR1: The **taper** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **description** of the **taper** shall be either 'angle taper', 'diameter taper', or 'directed taper'.

WR3: The **taper** shall have exactly one implicit representation defined by a **shape_representation_with_parameters.**

WR4: The **shape_representation_with_parameters** that represents the shape of the **taper** shall contain exactly one **representation_item** in its set of **items**.

WR5: If the **description** of the **taper** is 'angle taper', the implicit representation of the **taper** shall contain exactly one **representation_item** of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'taper angle'.

WR6: If the **description** of the **taper** is 'diameter taper', the implicit representation of the **taper** shall contain exactly one **representation_item** of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'final diameter'.

WR7: If the **description** of the **taper** is 'directed taper', the implicit representation of the **taper** shall contain exactly one **representation_item** of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'draft angle'.

WR8: If the **description** of the **taper** is 'directed taper', its shape shall be represented by exactly one **direction_shape_representation.** This **direction_shape_representation** shall have a **description** of 'draft direction'.

5.2.3.1.139 tee_profile

A **tee_profile** is a type of **shape_aspect** that consists of an upside down T shape and a reference system for the parameters of the **tee_profile.** The **tee_profile** consist of a stem and a perpendicular crossbar. The stem has two parallel lines connected at their ends by the crossbar. The crossbar is a rectangle. The profile is symmetric about the centreline of the stem. The

reference system is used to define location and orientation of the **tee_profile** and consists of two orthogonal reference axes X and Y that intersect in a reference point. See ARM definition for Tee_profile in paragraph 4.2.452 for more information.

EXPRESS specification:

```
*)
ENTITY tee_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)
         )) = 1;
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
           ({9 <= SIZEOF(pdr.used_representation.items) <= 10})
              )) = 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         NOT (srwp_i.name IN ['orientation','width','depth',
                                              'cross bar width','cross bar depth',
                                              'first offset','second offset',
                                              'first angle','second angle','radius'])
                       )) > 0)
              )) = 0;
  WR5:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'orientation') AND
                         ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i))
                       )) = 1)
              )) = 1;
  WR6:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'width') AND
                         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                     'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
```

```
          * TYPEOF(srwp_i)) = 2 )
                    )) = 1)
            )) = 1;
WR7:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
              (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                       (srwp_i.name = 'depth') AND
                       (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                  'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
          * TYPEOF(srwp_i)) = 2 )
                    )) = 1)
            )) = 1;
WR8:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
              (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                       (srwp_i.name = 'cross bar width') AND
         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                  'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
          * TYPEOF(srwp_i)) = 2 )
                    )) = 1)
            )) = 1;
WR9:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
              (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                       (srwp_i.name = 'cross bar depth') AND
         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                  'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
          * TYPEOF(srwp_i)) = 2 )
                    )) = 1)
            )) = 1;
WR10: SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
              (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                       (srwp_i.name = 'first offset') AND
         (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                  'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
          * TYPEOF(srwp_i)) = 2 )
                    )) = 1)
            )) = 1;
WR11: SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
              (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
```

```
                        (srwp_i.name = 'second offset') AND
          (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                   'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 )
                     )) = 1)
             )) = 1;
  WR12: SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
               (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                        (srwp_i.name = 'first angle') AND
          (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                   'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 )
                     )) = 1)
             )) = 1;
  WR13: SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
               (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                        (srwp_i.name = 'second angle') AND
          (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                   'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 )
                     )) = 1)
             )) = 1;
  WR14: SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) AND
               (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                        (srwp_i.name = 'radius') AND
          (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                   'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 )
                     )) = 1)
             )) <= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **tee_profile** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **tee_profile** shall have exactly one implicit representation defined by a **shape_representation_with_parameters.**

WR3: The **shape_representation_with_parameters** that represents the **tee_profile** shall contain nine or ten **representation_items** in its set of **items**.

WR4: The implicit representation of a **tee_profile** shall contain only **representation_items** in its set of **items** that have a **name** of either 'orientation', 'width', 'depth', 'cross bar width', 'cross bar depth', 'first offset', 'second offset', 'first angle', 'second angle', or 'radius'.

WR5: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **placement** with a **name** of 'orientation'.

WR6: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'width'.

WR7: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'depth'.

WR8: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'crossbar width'.

WR9: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'crossbar depth'.

WR10: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'first offset'.

WR11: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'second offset'.

WR12: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'first angle'.

WR13: Exactly one **representation_item** used for the implicit representation of a **tee_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'second angle'.

WR14: At most one **representation_item** used for the implicit representation of a **tee_profile** shall have a **name** of 'radius'. If present, this **representation_item** shall be of type **measure_representation_item** and **length_measure_with_unit.**

Informal propositions:

IP1: The **tee_profile** shall be planar and shall lie in the X-Y plane of the reference system.

IP2: The reference point of the **tee_profile** is defined at the midpoint of the side of the crossbar opposite the stem.

IP3: The **tee_profile** is defined with the side of the crossbar opposite the stem in the X direction and the profile symmetric with respect to the Y direction.

5.2.3.1.140 thread

A **thread** is a type of **feature_definition** that is the specification of a thread. See ARM definition for Thread_specification in paragraph 4.2.458 and for Defined_thread in paragraph 4.2.127 for more information.

EXPRESS specification:

```
*)
ENTITY thread
  SUBTYPE OF (feature_definition);
WHERE
  WR1:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                          'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
         (SIZEOF(pdr.used_representation.items) = 10)
               )) = 1
              )) = 1;
  WR2:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                          'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                  TYPEOF(pdr.used_representation)) AND
         (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                                 ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                                   TYPEOF(srwp_i)) AND
                                 (srwp_i.name = 'qualifier') )) = 1 )
               )) = 1
              )) = 1;
  WR3:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                          'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)) AND
         (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                  (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                           'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                        * TYPEOF(srwp_i)) = 2 ) AND
                                 (srwp_i.name = 'major diameter') )) = 1 )
               )) = 1
              )) = 1;
  WR4:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                          'PROPERTY_DEFINITION.DEFINITION') |
         SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
```

```
          TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                 (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                          'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                       * TYPEOF(srwp_i)) = 2 ) AND
                                (srwp_i.name = 'minor diameter') )) = 1 )
              )) = 1
             )) = 1;
WR5:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                        'PROPERTY_DEFINITION.DEFINITION') |
        SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                 (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                          'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                       * TYPEOF(srwp_i)) = 2 ) AND
                                (srwp_i.name = 'pitch diameter') )) = 1 )
              )) = 1
             )) = 1;
WR6:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                        'PROPERTY_DEFINITION.DEFINITION') |
        SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                 (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                          'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                       * TYPEOF(srwp_i)) = 2 ) AND
                                (srwp_i.name = 'crest') )) = 1 )
              )) = 1
             )) = 1;
WR7:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                        'PROPERTY_DEFINITION.DEFINITION') |
        SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                 (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                          'AUTOMOTIVE_DESIGN.RATIO_MEASURE_WITH_UNIT']
                       * TYPEOF(srwp_i)) = 2 ) AND
                                (srwp_i.name = 'density') )) = 1 )
              )) = 1
             )) = 1;
WR8:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                        'PROPERTY_DEFINITION.DEFINITION') |
        SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
```

```
          TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                                ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                                  TYPEOF(srwp_i)) AND
                                (srwp_i.name = 'fit class') )) = 1 )
             )) = 1
            )) = 1;
  WR9:  SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                          'PROPERTY_DEFINITION.DEFINITION') |
        SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                                ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                                  TYPEOF(srwp_i)) AND
                                (srwp_i.name = 'form') )) = 1 )
             )) = 1
            )) = 1;
  WR10: SIZEOF( QUERY( pd <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                          'PROPERTY_DEFINITION.DEFINITION') |
        SIZEOF( QUERY( pdr <* USEDIN( pd, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                                ('AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN
                                  TYPEOF(srwp_i)) AND
                                (srwp_i.name = 'hand') )) = 1 )
             )) = 1
            )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **shape_representation_with_parameters** that represents the **thread** shall contain exactly ten **representation_items** in its set of **items**.

WR2: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **descriptive_representation_item** with a **name** of 'qualifier'.

WR3: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'major diameter'.

WR4: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'minor diameter'.

WR5: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name**

of 'pitch diameter'.

WR6: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'crest'.

WR7: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **measure_representation_item** and **ratio_measure_with_unit** with a **name** of 'density'.

WR8: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **descriptive_representation_item** with a **name** of 'fit class'.

WR9: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **descriptive_representation_item** with a **name** of 'form'.

WR10: Exactly one **representation_item** used for the implicit representation of the **thread** shall be of type **descriptive_representation_item** with a **name** of 'hand'.

5.2.3.1.141 thread_occurrence

A **thread_occurrence** is a type of **shape_aspect** that is the representation of the bounded enclosed area on which a **thread** occurs. See ARM definition for Thread_feature in paragraph 4.2.457 for more information.

EXPRESS specification:

```
*)
ENTITY thread_occurrence
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  SIZEOF(['AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION',
                 'AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_SOLID']
                * TYPEOF(SELF.of_shape.definition)) = 1;
  WR2:  SELF.description IN ['inner','outer'];
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
           TYPEOF(pdr.used_representation)
               )) = 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
                 ({2 <= SIZEOF(pdr.used_representation.items) <= 3})
               )) = 1;
  WR5:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
                 (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
```

```
                        NOT (srwp_i.name IN ['orientation','effective length',
                                             'maximum length'])
                     )) > 0 )
             )) = 0;
  WR6:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                 (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                          'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                * TYPEOF(srwp_i)) = 2 ) AND
                        (srwp_i.name = 'effective length')
               )) = 1 )
             )) = 1;
  WR7:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                 ('AUTOMOTIVE_DESIGN.PLACEMENT' IN
                   TYPEOF(srwp_i)) AND
                        (srwp_i.name = 'orientation')
               )) = 1 )
             )) = 1;
  WR8:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
        (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                 (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                          'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT']
                                * TYPEOF(srwp_i)) = 2 ) AND
                        (srwp_i.name = 'maximum length')
               )) <= 1 )
             )) = 1;
  WR9:  SIZEOF( QUERY( sar <* USEDIN(SELF, 'AUTOMOTIVE_DESIGN.' +
                            'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
               SIZEOF(['AUTOMOTIVE_DESIGN.THREAD',
                       'AUTOMOTIVE_DESIGN.EXTERNALLY_DEFINED_THREAD']
                     * TYPEOF(sar.relating_shape_aspect.of_shape.definition)) = 1
             )) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **thread_occurrence** shall be an aspect of the shape of either a **product_definition** or a **compound_feature_in_solid.**

WR2: The **thread_occurrence** shall have a description of either 'inner' or 'outer'.

WR3: The **thread_occurence** shall have its implicit representation specified by exactly one **shape_representation_with_parameters.**

WR4: The implicit representation of the **thread_occurrence** shall contain two or three **representation_items** in its set of **items**.

WR5: The implicit representation of a **thread_occurrence** shall contain only **representation_items** in its set of **items** that have a **name** of either 'orientation', 'effective length', or 'maximum length'.

WR6: Exactly one **representation_item** used for the implicit representation of the **thread_occurence** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'effective length'.

WR7: Exactly one **representation_item** used for the implicit representation of the **thread_occurence** shall be of type **placement** with a **name** of 'orientation'.

WR8: At most one **representation_item** used for the implicit representation of the **thread_occurence** shall have a name of 'maximum length'. If present, this **representation_item** shall be of type **measure_representation_item** and **length_measure_with_unit.**

WR9: Each **thread_occurrence** shall be referenced as **related_shape_aspect** by a **shape_aspect_relationship** which has a **relating_shape_aspect** that is an aspect of the shape of a **thread** or of an **externally_defined_thread.**

5.2.3.1.142 transition_feature

A **transition_feature** is a type of **shape_aspect** that is the representation of a transition between two **face_surfaces.** See ARM definition for Transition_feature in paragraph 4.2.476 for more information.

EXPRESS specification:

```
*)
ENTITY transition_feature
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  SIZEOF(['AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION',
         'AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_PANEL',
         'AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_SOLID']
               * TYPEOF(SELF.of_shape.definition)) = 1;
  WR2:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION' IN
           TYPEOF(pdr.used_representation)) AND
         (pdr.used_representation.description = 'first face shape')
              )) = 1;
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION' IN
           TYPEOF(pdr.used_representation)) AND
```

```
        (pdr.used_representation.description = 'second face shape')
             )) = 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ('AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION' IN
          TYPEOF(pdr.used_representation)) AND
        (pdr.used_representation.description = 'transition face shape')
             )) = 1;
  WR5:  SIZEOF(['AUTOMOTIVE_DESIGN.CHAMFER',
        'AUTOMOTIVE_DESIGN.EDGE_ROUND',
        'AUTOMOTIVE_DESIGN.FILLET']
              * TYPEOF(SELF)) = 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **transition_feature** shall be an aspect of the shape of either a **product_definition**, a **compound_feature_in_panel**, or a **compound_feature_in_solid.**

WR2: The **transition_feature** shall have exactly one **face_shape_representation** with a **description** of 'first face shape'.

WR3: The **transition_feature** shall have exactly one **face_shape_representation** with a **description** of 'second face shape'.

WR4: The **transition_feature** shall have exactly one **face_shape_representation** with a **description** of 'transition face shape'.

WR5: Each instance of **transition_feature** shall be either a **chamfer**, an **edge_round,** or a **fillet**.

5.2.3.1.143 vector_style

A **vector_style** is a type of **curve_style** and **pre_defined_terminator_symbol** that defines the visual appearance of a vector.

EXPRESS specification:

```
*)
ENTITY vector_style
  SUBTYPE OF (curve_style, pre_defined_terminator_symbol);
END_ENTITY;
(*
```

5.2.3.1.144 vee_profile

A **vee_profile** is a type of **shape_aspect** that consists of two connected straight lines and a reference system for the parameters of the **vee_profile.** The reference system consists of two

orthogonal reference axes X and Y that intersect in a reference point. See ARM definition for Vee_profile in paragraph 4.2.487 for more information.

EXPRESS specification:

```
*)
ENTITY vee_profile
  SUBTYPE OF (shape_aspect);
WHERE
  WR1:  'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN
         TYPEOF(SELF.of_shape.definition);
  WR2:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)
             )) = 1;
  WR3:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
           ({3 <= SIZEOF(pdr.used_representation.items) <= 4})
             )) = 1;
  WR4:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         NOT (srwp_i.name IN ['orientation','profile angle',
                                              'tilt angle','profile radius'])
                        )) > 0)
             )) = 0;
  WR5:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'orientation') AND
                         ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i))
                        )) = 1)
             )) = 1;
  WR6:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'profile angle') AND
          (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                   'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 )
                        )) = 1)
             )) = 1;
```

```
  WR7:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'tilt angle') AND
          (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
           'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 )
                      )) = 1)
             )) = 1;
  WR8:  SIZEOF( QUERY( pdr <* USEDIN( SELF, 'AUTOMOTIVE_DESIGN.' +
                      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
          TYPEOF(pdr.used_representation)) AND
                (SIZEOF( QUERY( srwp_i <* pdr.used_representation.items |
                         (srwp_i.name = 'profile radius') AND
          (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
                   'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT']
           * TYPEOF(srwp_i)) = 2 )
                      )) = 1)
             )) <= 1;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **vee_profile** shall be an aspect of the shape of a **feature_component_definition.**

WR2: The **vee_profile** shall have exactly one implicit representation defined by a **shape_representation_with_parameters.**

WR3: The **shape_representation_with_parameters** that represents the **vee_profile** shall contain three or four **representation_items** in its set of **items**.

WR4: The implicit representation of a **vee_profile** shall contain only **representation_items** in its set of **items** that have a **name** of either 'orientation', 'profile angle', 'tilt angle', or 'profile radius'.

WR5: Exactly one **representation_item** used for the implicit representation of a **vee_profile** shall be of type **placement** with a **name** of 'orientation'.

WR6: Exactly one **representation_item** used for the implicit representation of a **vee_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'profile angle'.

WR7: Exactly one **representation_item** used for the implicit representation of a **vee_profile** shall be of type **measure_representation_item** and **plane_angle_measure_with_unit** with a **name** of 'tilt angle'.

WR8: At most one **representation_item** used for the implicit representation of a **vee_profile** shall have a **name** of 'profile radius'. If present, this **representation_item** shall be of type

measure_representation_item with a **value_component** of type **length_measure_with_-unit.**

Informal propositions:

IP1: The **vee_profile** shall be planar and shall lie in the X-Y plane of the reference system.

IP2: The reference point of the **vee_profile** is defined at the position where the two lines connect.

5.2.3.1.145 version_assignment

A **version_assignment** is an **identification_assignment** that assigns a version identifier to a **version_item.**

EXPRESS specification:

```
*)
ENTITY version_assignment
  SUBTYPE OF (identification_assignment);
  items : SET [1:?] OF version_item;
WHERE
  WR1:  SELF.role.name = 'version id';
  WR2:  SIZEOF(SELF.items) = 1;
END_ENTITY;
(*
```

Attribute definitions:

items: The set of items to which the version identifier is assigned.

Formal propositions:

WR1: The **version_assignment** shall reference as its **role** an **identification_role** that has a **name** of 'version id'.

WR2: The **version_assignment** shall contain exactly one item in its set of **items**.

5.2.3.2 automotive_design imported entity modifications

5.2.3.2.1 action

The base definition of the **action** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **action** entity:

– action_subtype_restriction (see 5.2.4.2);

– activity_requires_person_or_organization_assignment (see 5.2.4.3);

– restrict_action_resource_requirement_for_process_operation (see 5.2.4.73);

– restrict_multi_language_for_action (see 5.2.4.79).

5.2.3.2.2 action_assignment

The base definition of the **action_assignment** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **action_assignment** entity:

– action_assignment_subtype_exclusiveness (see 5.2.4.1).

5.2.3.2.3 action_directive

The base definition of the **action_directive** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **action_directive** entity:

– restrict_action_directive (see 5.2.4.72);

– restrict_multi_language_for_action_directive (see 5.2.4.80);

– restrict_version_assignment_for_action_directive (see 5.2.4.168).

5.2.3.2.4 action_method

The base definition of the **action_method** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **action_method** entity:

– restrict_multi_language_for_action_method (see 5.2.4.81).

5.2.3.2.5 action_property

The base definition of the **action_property** entity is given in ISO 10303–49. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **action_property** entity:

– restrict_multi_language_for_action_property (see 5.2.4.82).

5.2.3.2.6 action_relationship

The base definition of the **action_relationship** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **action_relationship** entity:

– restrict_multi_language_for_action_relationship (see 5.2.4.83).

5.2.3.2.7 action_request_role

The base definition of the **action_request_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **action_request_role** entity:

– dependent_instantiable_action_request_role (see 5.2.4.12).

5.2.3.2.8 action_resource_requirement

The base definition of the **action_resource_requirement** entity is given in ISO 10303–49. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **action_resource_requirement** entity:

– restrict_action_resource_requirement_for_process_operation (see 5.2.4.73);

– restrict_multi_language_for_action_resource_requirement (see 5.2.4.84).

5.2.3.2.9 action_role

The base definition of the **action_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **action_role** entity:

– dependent_instantiable_action_role (see 5.2.4.13).

5.2.3.2.10 alternate_product_relationship

The base definition of the **alternate_product_relationship** entity is given in ISO 10303–44. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **alternate_product_relationship** entity:

– restrict_multi_language_for_alternate_product_relationship (see 5.2.4.85).

5.2.3.2.11 application_context

The base definition of the **application_context** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **application_context** entity:

– application_context_requires_ap_definition (see 5.2.4.4);

– restrict_multi_language_for_application_context (see 5.2.4.86).

5.2.3.2.12 approval

The base definition of the **approval** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **approval** entity:

– approval_requires_approval_assignment (see 5.2.4.6);

– restrict_approval_date_time_for_approval (see 5.2.4.74);

– restrict_multi_language_for_approval (see 5.2.4.87).

5.2.3.2.13 approval_date_time_role

The base definition of the **approval_date_time_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **approval_date_time_role** entity:

– dependent_instantiable_approval_date_time_role (see 5.2.4.14);

– restrict_approval_date_time_role (see 5.2.4.75).

5.2.3.2.14 approval_person_organization

The base definition of the **approval_person_organization** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **approval_person_organization** entity:

– approval_person_organization_requires_date_time (see 5.2.4.5).

5.2.3.2.15 approval_role

The base definition of the **approval_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **approval_role** entity:

– dependent_instantiable_approval_role (see 5.2.4.15).

5.2.3.2.16 approval_status

The base definition of the **approval_status** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **approval_status** entity:

– restrict_classification_system_for_approval_status (see 5.2.4.76).

5.2.3.2.17 approver_role

The base definition of the **approver_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **approver_role** entity:

– dependent_instantiable_approver_role (see 5.2.4.16).

5.2.3.2.18 assembly_component_usage

The base definition of the **assembly_component_usage** entity is given in ISO 10303–44. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **assembly_component_usage** entity:

– restrict_multi_language_for_assembly_component_usage (see 5.2.4.88).

5.2.3.2.19 assembly_component_usage_substitute

The base definition of the **assembly_component_usage_substitute** entity is given in ISO 10303–44. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **assembly_component_usage_substitute** entity:

– restrict_multi_language_for_assembly_component_usage_substitute (see 5.2.4.89).

5.2.3.2.20 camera_image

The base definition of the **camera_image** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **camera_image** entity:

– subtype_mandatory_camera_image_2d_or_3d (see 5.2.4.190).

5.2.3.2.21 camera_image_3d_with_scale

The base definition of the **camera_image_3d_with_scale** entity is given in ISO 10303–517. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **camera_image_3d_with_scale** entity:

– restrict_multi_language_for_camera_image_3d_with_scale (see 5.2.4.91).

5.2.3.2.22 camera_model

The base definition of the **camera_model** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **camera_model** entity:

– subtype_mandatory_camera_model_2d_or_3d (see 5.2.4.191).

5.2.3.2.23 certification

The base definition of the **certification** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **certification** entity:

– restrict_multi_language_for_certification (see 5.2.4.92).

5.2.3.2.24 certification_role

The base definition of the **certification_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **certification_role** entity:

– dependent_instantiable_certification_role (see 5.2.4.17).

5.2.3.2.25 characterized_object

The base definition of the **characterized_object** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **characterized_object** entity:

– characterized_object_subtype_exclusiveness (see 5.2.4.7).

5.2.3.2.26 colour_rgb

The base definition of the **colour_rgb** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **colour_rgb** entity:

– dependent_instantiable_colour_rgb (see 5.2.4.18).

5.2.3.2.27 configurable_item

The base definition of the **configurable_item** entity is given in ISO 10303–44. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **configurable_item** entity:

– restrict_multi_language_for_configurable_item (see 5.2.4.95);

– restrict_product_concept_features_for_product_specification (see 5.2.4.167);

– restrict_version_assignment_for_configurable_item (see 5.2.4.169).

5.2.3.2.28 configuration_design

The base definition of the **configuration_design** entity is given in ISO 10303–44. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **configuration_design** entity:

– configuration_design_subtype_exclusiveness (see 5.2.4.8);

– restrict_multi_language_for_configuration_design (see 5.2.4.96).

5.2.3.2.29 configuration_item_relationship

The base definition of the **configuration_item_relationship** entity is given in ISO 10303–44. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **configuration_item_relationship** entity:

– restrict_multi_language_for_configuration_item_relationship (see 5.2.4.97).

5.2.3.2.30 contract_role

The base definition of the **contract_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **contract_role** entity:

– dependent_instantiable_contract_role (see 5.2.4.19).

5.2.3.2.31 curve_style

The base definition of the **curve_style** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **curve_style** entity:

– dependent_instantiable_curve_style (see 5.2.4.20).

5.2.3.2.32 curve_style_font

The base definition of the **curve_style_font** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **curve_style_font** entity:

– curve_font_usage (see 5.2.4.10).

5.2.3.2.33 data_environment

The base definition of the **data_environment** entity is given in ISO 10303–45. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **data_environment** entity:

– restrict_multi_language_for_data_environment (see 5.2.4.99).

5.2.3.2.34 date

The base definition of the **date** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **date** entity:

– dependent_instantiable_date (see 5.2.4.21).

5.2.3.2.35 date_and_time

The base definition of the **date_and_time** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **date_and_time** entity:

– dependent_instantiable_date_and_time (see 5.2.4.22).

5.2.3.2.36 date_role

The base definition of the **date_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **date_role** entity:

– dependent_instantiable_date_role (see 5.2.4.23).

5.2.3.2.37 date_time_role

The base definition of the **date_time_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **date_time_role** entity:

– dependent_instantiable_date_time_role (see 5.2.4.24).

5.2.3.2.38 derived_unit

The base definition of the **derived_unit** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **derived_unit** entity:

– dependent_instantiable_derived_unit (see 5.2.4.25).

5.2.3.2.39 descriptive_representation_item

The base definition of the **descriptive_representation_item** entity is given in ISO 10303–45. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **descriptive_representation_item** entity:

– restrict_multi_language_for_descriptive_representation_item (see 5.2.4.101).

5.2.3.2.40 dimensional_location

The base definition of the **dimensional_location** entity is given in ISO 10303–47. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **dimensional_location** entity:

– restrict_multi_language_for_dimensional_location (see 5.2.4.102).

5.2.3.2.41 dimensional_size

The base definition of the **dimensional_size** entity is given in ISO 10303–47. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **dimensional_size** entity:

– dependent_instantiable_dimensional_size (see 5.2.4.26);

– restrict_multi_language_for_dimensional_size (see 5.2.4.103).

5.2.3.2.42 document

The base definition of the **document** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **document** entity:

– restrict_multi_language_for_document (see 5.2.4.104);

– restrict_version_assignment_for_document (see 5.2.4.170).

5.2.3.2.43 document_relationship

The base definition of the **document_relationship** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **document_relationship** entity:

– restrict_multi_language_for_document_relationship (see 5.2.4.105).

5.2.3.2.44 document_role

The base definition of the **document_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **document_role** entity:

– dependent_instantiable_document_role (see 5.2.4.27).

5.2.3.2.45 document_type

The base definition of the **document_type** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **document_type** entity:

– restrict_classification_system_for_document_type (see 5.2.4.77).

5.2.3.2.46 draughting_subfigure_representation

The base definition of the **draughting_subfigure_representation** entity is given in ISO 10303–504. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **draughting_subfigure_representation** entity:

– draughting_subfigure_representation_layers (see 5.2.4.52).

5.2.3.2.47 draughting_title

The base definition of the **draughting_title** entity is given in ISO 10303–101. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **draughting_title** entity:

– restrict_multi_language_for_draughting_title (see 5.2.4.107).

5.2.3.2.48 drawing_revision

The base definition of the **drawing_revision** entity is given in ISO 10303–101. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **drawing_revision** entity:

– subtype_mandatory_drawing_revision (see 5.2.4.192).

5.2.3.2.49 drawing_sheet_revision

The base definition of the **drawing_sheet_revision** entity is given in ISO 10303–101. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **drawing_sheet_-revision** entity:

– drawing_sheet_annotation_layers (see 5.2.4.53);

– restrict_multi_language_for_drawing_sheet_revision (see 5.2.4.108);

– sheets_belong_to_one_drawing (see 5.2.4.188).

5.2.3.2.50 drawing_sheet_revision_usage

The base definition of the **drawing_sheet_revision_usage** entity is given in ISO 10303–101. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **drawing_sheet_-revision_usage** entity:

– sheets_belong_to_one_drawing (see 5.2.4.188).

5.2.3.2.51 effective_context_assignment

The base definition of the **effective_context_assignment** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **effective_context_-assignment** entity:

– effective_context_assignment_subtype_exclusiveness (see 5.2.4.56).

5.2.3.2.52 effective_item_assignment

The base definition of the **effective_item_assignment** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **effective_item_-assignment** entity:

– effective_item_assignment_subtype_exclusiveness (see 5.2.4.57).

5.2.3.2.53 effectivity

The base definition of the **effectivity** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **effectivity** entity:

– effectivity_requires_effective_item_assignment (see 5.2.4.58);

– restrict_multi_language_for_effectivity (see 5.2.4.109);

– subtype_mandatory_effectivity (see 5.2.4.193).

5.2.3.2.54 effectivity_relationship

The base definition of the **effectivity_relationship** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **effectivity_relationship** entity:

– restrict_multi_language_for_effectivity_relationship (see 5.2.4.110).

5.2.3.2.55 effectivity_role

The base definition of the **effectivity_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **effectivity_role** entity:

– dependent_instantiable_effectivity_role (see 5.2.4.28).

5.2.3.2.56 externally_defined_curve_font

The base definition of the **externally_defined_curve_font** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **externally_defined_-curve_font** entity:

– curve_font_usage (see 5.2.4.10).

5.2.3.2.57 externally_defined_symbol

The base definition of the **externally_defined_symbol** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **externally_defined_-symbol** entity:

– dependent_instantiable_externally_defined_symbol (see 5.2.4.29).

5.2.3.2.58 externally_defined_text_font

The base definition of the **externally_defined_text_font** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **externally_defined_text_font** entity:

– text_font_usage (see 5.2.4.200).

5.2.3.2.59 fill_area_style

The base definition of the **fill_area_style** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **fill_area_style** entity:

– dependent_instantiable_fill_area_style (see 5.2.4.30).

5.2.3.2.60 fill_area_style_colour

The base definition of the **fill_area_style_colour** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **fill_area_style_colour** entity:

– dependent_instantiable_fill_area_style_colour (see 5.2.4.31).

5.2.3.2.61 fill_area_style_tile_symbol_with_style

The base definition of the **fill_area_style_tile_symbol_with_style** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **fill_area_style_tile_symbol_with_style** entity:

– fill_area_style_tile_symbol_constraint (see 5.2.4.59).

5.2.3.2.62 geometric_representation_context

The base definition of the **geometric_representation_context** entity is given in ISO 10303–42. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **geometric_representation_-context** entity:

– dimensionality_is_two_or_three (see 5.2.4.50).

5.2.3.2.63 geometric_tolerance

The base definition of the **geometric_tolerance** entity is given in ISO 10303–47. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **geometric_tolerance** entity:

– geometric_tolerance_subtype_exclusiveness (see 5.2.4.61);

– restrict_multi_language_for_geometric_tolerance (see 5.2.4.115).

5.2.3.2.64 global_unit_assigned_context

The base definition of the **global_unit_assigned_context** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **global_unit_assigned_-context** entity:

– global_length_and_angle_units_2d_or_3d (see 5.2.4.62).

5.2.3.2.65 group

The base definition of the **group** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **group** entity:

– restrict_multi_language_for_group (see 5.2.4.116).

5.2.3.2.66 group_assignment

The base definition of the **group_assignment** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **group_assignment** entity:

– group_assignment_subtype_exclusiveness (see 5.2.4.63).

5.2.3.2.67 group_role

The base definition of the **group_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **group_role** entity:

– dependent_instantiable_group_role (see 5.2.4.32).

5.2.3.2.68 identification_assignment

The base definition of the **identification_assignment** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **identification_assignment** entity:

– identification_assignment_subtype_exclusiveness (see 5.2.4.64).

5.2.3.2.69 identification_role

The base definition of the **identification_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **identification_role** entity:

– dependent_instantiable_identification_role (see 5.2.4.33);

– restrict_multi_language_for_identification_role (see 5.2.4.117).

5.2.3.2.70 language_role

The base definition of the **language_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **language_role** entity:

– dependent_instantiable_language_role (see 5.2.4.34).

5.2.3.2.71 make_from_usage_option

The base definition of the **make_from_usage_option** entity is given in ISO 10303–44. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **make_from_usage_-option** entity:

– restrict_multi_language_for_make_from_usage_option (see 5.2.4.118).

5.2.3.2.72 mapped_item

The base definition of the **mapped_item** entity is given in ISO 10303–43. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **mapped_item** entity:

– drawing_sheet_layout_usage (see 5.2.4.54).

5.2.3.2.73 material_property

The base definition of the **material_property** entity is given in ISO 10303–45. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **material_property** entity:

– restrict_multi_language_for_material_property (see 5.2.4.119).

5.2.3.2.74 measure_representation_item

The base definition of the **measure_representation_item** entity is given in ISO 10303–45. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **measure_representation_-item** entity:

– restrict_multi_language_for_measure_representation_item (see 5.2.4.120).

5.2.3.2.75 measure_with_unit

The base definition of the **measure_with_unit** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **measure_with_unit** entity:

– dependent_instantiable_measure_with_unit (see 5.2.4.35).

5.2.3.2.76 mechanical_design_geometric_presentation_representation

The base definition of the **mechanical_design_geometric_presentation_representation** entity is given in ISO 10303–517. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **mechanical_design_geometric_presentation_representation** entity:

– restrict_multi_language_for_mechanical_design_geometric_presentation_representation (see 5.2.4.121);

– restrict_version_assignment_for_mechanical_design_geometric_presentation_representation (see 5.2.4.172).

5.2.3.2.77 named_unit

The base definition of the **named_unit** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **named_unit** entity:

– dependent_instantiable_named_unit (see 5.2.4.36).

5.2.3.2.78 next_assembly_usage_occurrence

The base definition of the **next_assembly_usage_occurrence** entity is given in ISO 10303–44. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **next_assembly_usage_occurrence** entity:

– coordinated_assembly_and_shape (see 5.2.4.9).

5.2.3.2.79 organization_assignment

The base definition of the **organization_assignment** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **organization_assignment** entity:

– organization_assignment_subtype_exclusiveness (see 5.2.4.65).

5.2.3.2.80 organization_relationship

The base definition of the **organization_relationship** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **organization_relationship** entity:

– restrict_multi_language_for_organization_relationship (see 5.2.4.123).

5.2.3.2.81 organization_role

The base definition of the **organization_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **organization_role** entity:

– dependent_instantiable_organization_role (see 5.2.4.37);

– restrict_multi_language_for_organization_role (see 5.2.4.124).

5.2.3.2.82 pair_actuator

The base definition of the **pair_actuator** entity is given in ISO 10303–105. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **pair_actuator** entity:

– restrict_multi_language_for_pair_actuator (see 5.2.4.125).

5.2.3.2.83 person

The base definition of the **person** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **person** entity:

– person_requires_person_and_organization (see 5.2.4.66).

5.2.3.2.84 person_and_organization_role

The base definition of the **person_and_organization_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **person_and_organization_-role** entity:

– dependent_instantiable_person_and_organization_role (see 5.2.4.38).

5.2.3.2.85 person_role

The base definition of the **person_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **person_role** entity:

– dependent_instantiable_person_role (see 5.2.4.39).

5.2.3.2.86 pre_defined_colour

The base definition of the **pre_defined_colour** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **pre_defined_colour** entity:

– dependent_instantiable_pre_defined_colour (see 5.2.4.40);

– subtype_mandatory_pre_defined_colour (see 5.2.4.194).

5.2.3.2.87 pre_defined_curve_font

The base definition of the **pre_defined_curve_font** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **pre_defined_curve_-font** entity:

– curve_font_usage (see 5.2.4.10);

– subtype_mandatory_pre_defined_curve_font (see 5.2.4.195).

5.2.3.2.88 pre_defined_symbol

The base definition of the **pre_defined_symbol** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **pre_defined_symbol** entity:

– dependent_instantiable_pre_defined_symbol (see 5.2.4.41).

5.2.3.2.89 pre_defined_text_font

The base definition of the **pre_defined_text_font** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **pre_defined_text_font** entity:

– subtype_mandatory_pre_defined_text_font (see 5.2.4.196);

– text_font_usage (see 5.2.4.200).

5.2.3.2.90 presentation_area

The base definition of the **presentation_area** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **presentation_area** entity:

– restrict_multi_language_for_presentation_area (see 5.2.4.126);

– restrict_version_assignment_for_presentation_area (see 5.2.4.174).

5.2.3.2.91 presentation_layer_assignment

The base definition of the **presentation_layer_assignment** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **presentation_layer_-assignment** entity:

– presentation_layer_assignment_constraint_2d_or_3d (see 5.2.4.67);

– restrict_multi_language_for_presentation_layer_assignment (see 5.2.4.127).

5.2.3.2.92 presentation_style_by_context

The base definition of the **presentation_style_by_context** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **presentation_style_-by_context** entity:

– dependent_instantiable_presentation_style_by_context (see 5.2.4.42).

5.2.3.2.93 presentation_view

The base definition of the **presentation_view** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **presentation_view** entity:

– drawing_view_annotation_layers (see 5.2.4.55);

– restrict_multi_language_for_presentation_view (see 5.2.4.128).

5.2.3.2.94 process_product_association

The base definition of the **process_product_association** entity is given in ISO 10303–49. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **process_product_-association** entity:

– restrict_multi_language_for_process_product_association (see 5.2.4.132);

– restrict_process_product_association (see 5.2.4.165).

5.2.3.2.95 product

The base definition of the **product** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product** entity:

– product_requires_version (see 5.2.4.70);

– restrict_multi_language_for_product (see 5.2.4.133).

5.2.3.2.96 product_category

The base definition of the **product_category** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **product_category** entity:

– restrict_multi_language_for_product_category (see 5.2.4.134).

5.2.3.2.97 product_concept

The base definition of the **product_concept** entity is given in ISO 10303–44. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **product_concept** entity:

– subtype_mandatory_product_concept (see 5.2.4.197).

5.2.3.2.98 product_concept_feature

The base definition of the **product_concept_feature** entity is given in ISO 10303–44. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_concept_-feature** entity:

– restrict_multi_language_for_product_concept_feature (see 5.2.4.137);

– restrict_version_assignment_for_product_concept_feature (see 5.2.4.178).

5.2.3.2.99 product_concept_feature_association

The base definition of the **product_concept_feature_association** entity is given in ISO 10303–44. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **product_concept_-feature_association** entity:

– restrict_multi_language_for_product_concept_feature_association (see 5.2.4.138).

5.2.3.2.100 product_definition

The base definition of the **product_definition** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_definition** entity:

– functional_product_definition_requires_functional_product (see 5.2.4.60);

– product_definition_subtype_exclusiveness (see 5.2.4.69);

– restrict_multi_language_for_product_definition (see 5.2.4.140);

– restrict_version_assignment_for_product_definition (see 5.2.4.179).

5.2.3.2.101 product_definition_formation

The base definition of the **product_definition_formation** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **product_definition_formation** entity:

– restrict_multi_language_for_product_definition_formation (see 5.2.4.141).

5.2.3.2.102 product_definition_formation_relationship

The base definition of the **product_definition_formation_relationship** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **product_definition_formation_relationship** entity:

– restrict_multi_language_for_product_definition_formation_relationship (see 5.2.4.142).

5.2.3.2.103 product_definition_relationship

The base definition of the **product_definition_relationship** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_definition_relationship** entity:

– definition_replacement_requires_effectivity_assignment (see 5.2.4.11);

– restrict_multi_language_for_product_definition_relationship (see 5.2.4.144);

– restrict_product_categories_for_tool_part_relationship (see 5.2.4.166).

5.2.3.2.104 product_definition_shape

The base definition of the **product_definition_shape** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **product_definition_shape** entity:

– restrict_multi_language_for_product_definition_shape (see 5.2.4.145).

5.2.3.2.105 product_definition_substitute

The base definition of the **product_definition_substitute** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_definition_substitute** entity:

– product_definition_substitute_requires_effectivity_assignment (see 5.2.4.68);

– restrict_multi_language_for_product_definition_substitute (see 5.2.4.146).

5.2.3.2.106 product_definition_with_associated_documents

The base definition of the **product_definition_with_associated_documents** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **product_definition_with_associated_documents** entity:

– subtype_mandatory_product_definition_with_associated_documents (see 5.2.4.198).

5.2.3.2.107 product_related_product_category

The base definition of the **product_related_product_category** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **product_related_product_category** entity:

– restrict_product_categories_for_tool_part_relationship (see 5.2.4.166).

5.2.3.2.108 property_definition

The base definition of the **property_definition** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **property_definition** entity:

– restrict_multi_language_for_property_definition (see 5.2.4.149).

5.2.3.2.109 property_definition_relationship

The base definition of the **property_definition_relationship** entity is given in ISO 10303–45. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **property_definition_-relationship** entity:

– restrict_multi_language_for_property_definition_relationship (see 5.2.4.150).

5.2.3.2.110 property_definition_representation

The base definition of the **property_definition_representation** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **property_definition_-representation** entity:

– restrict_multi_language_for_property_definition_representation (see 5.2.4.151).

5.2.3.2.111 representation

The base definition of the **representation** entity is given in ISO 10303–43. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **representation** entity:

– dependent_instantiable_representation (see 5.2.4.43);

– representation_subtype_exclusiveness (see 5.2.4.71);

– restrict_multi_language_for_representation (see 5.2.4.153);

– restrict_version_assignment_for_representation (see 5.2.4.180).

5.2.3.2.112 representation_item

The base definition of the **representation_item** entity is given in ISO 10303–43. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **representation_item** entity:

– restrict_multi_language_for_representation_item (see 5.2.4.154).

5.2.3.2.113 representation_relationship

The base definition of the **representation_relationship** entity is given in ISO 10303–43. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **representation_relationship** entity:

– restrict_multi_language_for_representation_relationship (see 5.2.4.155).

5.2.3.2.114 resource_requirement_type

The base definition of the **resource_requirement_type** entity is given in ISO 10303–49. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **resource_requirement_type** entity:

– restrict_multi_language_for_resource_requirement_type (see 5.2.4.156).

5.2.3.2.115 security_classification

The base definition of the **security_classification** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **security_classification** entity:

– restrict_multi_language_for_security_classification (see 5.2.4.158);

– security_classification_requires_security_classification_assignment (see 5.2.4.184).

5.2.3.2.116 security_classification_level

The base definition of the **security_classification_level** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **security_classification_-level** entity:

– restrict_classification_system_for_security_classification_level (see 5.2.4.78).

5.2.3.2.117 security_classification_role

The base definition of the **security_classification_role** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **security_classification_-role** entity:

– dependent_instantiable_security_classification_role (see 5.2.4.44).

5.2.3.2.118 shape_aspect

The base definition of the **shape_aspect** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **shape_aspect** entity:

– restrict_multi_language_for_shape_aspect (see 5.2.4.159);

– shape_aspect_subtype_exclusiveness (see 5.2.4.186).

5.2.3.2.119 shape_aspect_relationship

The base definition of the **shape_aspect_relationship** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **shape_aspect_-relationship** entity:

– restrict_multi_language_for_shape_aspect_relationship (see 5.2.4.160);

– shape_aspect_relationship_subtype_exclusiveness (see 5.2.4.185).

5.2.3.2.120 shape_dimension_representation

The base definition of the **shape_dimension_representation** entity is given in ISO 10303–47. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **shape_dimension_representation** entity:

– restrict_multi_language_for_shape_dimension_representation (see 5.2.4.161).

5.2.3.2.121 shape_representation

The base definition of the **shape_representation** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **shape_representation** entity:

– restrict_multi_language_for_shape_representation (see 5.2.4.162);

– restrict_version_assignment_for_shape_representation (see 5.2.4.181);

– shape_representation_subtype_exclusiveness (see 5.2.4.187).

5.2.3.2.122 styled_item

The base definition of the **styled_item** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **styled_item** entity:

– styled_curve (see 5.2.4.189).

5.2.3.2.123 symbol_colour

The base definition of the **symbol_colour** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **symbol_colour** entity:

– dependent_instantiable_symbol_colour (see 5.2.4.45).

5.2.3.2.124 terminator_symbol

The base definition of the **terminator_symbol** entity is given in ISO 10303–101. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **terminator_symbol** entity:

– terminator_symbol_constraint (see 5.2.4.199).

5.2.3.2.125 text_style

The base definition of the **text_style** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **text_style** entity:

– dependent_instantiable_text_style (see 5.2.4.46).

5.2.3.2.126 text_style_for_defined_font

The base definition of the **text_style_for_defined_font** entity is given in ISO 10303–46. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **text_style_for_-defined_font** entity:

– dependent_instantiable_text_style_for_defined_font (see 5.2.4.47).

5.2.3.2.127 tolerance_value

The base definition of the **tolerance_value** entity is given in ISO 10303–47. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **tolerance_value** entity:

– dependent_instantiable_tolerance_value (see 5.2.4.48).

5.2.3.2.128 uncertainty_qualifier

The base definition of the **uncertainty_qualifier** entity is given in ISO 10303–45. The following modifications apply to this part of ISO 10303.

Associated global rule:

The following global rule defined in this part of ISO 10303 applies to the **uncertainty_qualifier** entity:

– dependent_instantiable_uncertainty_qualifier (see 5.2.4.49).

5.2.3.2.129 versioned_action_request

The base definition of the **versioned_action_request** entity is given in ISO 10303–41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **versioned_action_request** entity:

– restrict_multi_language_for_versioned_action_request (see 5.2.4.164);

– versioned_action_request_requires_date_and_person_or_organization (see 5.2.4.201).

5.2.4 automotive_design rule definitions

5.2.4.1 action_assignment_subtype_exclusiveness

The **action_assignment_subtype_exclusiveness** rule specifies that an instance of the subtypes of **action_assignment** shall be one of **applied_action_assignment**, **event_assignment**, **event_context_assignment**, or **retention_assignment.**

EXPRESS specification:

```
*)
RULE action_assignment_subtype_exclusiveness FOR
    (action_assignment);
WHERE
  WR1:  SIZEOF (QUERY (aa <* action_assignment |
                (SIZEOF (TYPEOF (aa) *
                         ['AUTOMOTIVE_DESIGN.APPLIED_ACTION_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.ELEMENT_DELIVERY_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.EVENT_CONTEXT_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.PROCESS_PLAN_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.RENTENTION_ASSIGNMENT'])
                 > 1) OR
                ((SIZEOF (TYPEOF (aa)) > 2) AND
                 (SIZEOF (TYPEOF (aa) *
                         ['AUTOMOTIVE_DESIGN.APPLIED_ACTION_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.ELEMENT_DELIVERY_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.EVENT_CONTEXT_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.PROCESS_PLAN_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.RENTENTION_ASSIGNMENT'])
```

```
                    = 1))
        )) = 0;

END_RULE;
(*
```

Argument definitions:

action_assignment: the set of all instances of **action_assignment**.

Formal propositions:

WR1: Each instance of the subtypes of **action_assignment** shall be one of **applied_action_assignment**, **event_assignment**, **event_context_assignment**, or **retention_assignment**.

5.2.4.2 action_subtype_restriction

The **action_subtype_restriction** rule specifies that an instance of the subtypes of action shall be either one of **executed_action**, **event_reference**, **retention**, **property_process**, or **product_definition_process**, or shall be a complex instance of **property_process** and **product_definition_process**.

EXPRESS specification:

```
*)
RULE action_subtype_restriction FOR
    (action);
WHERE
  WR1:  SIZEOF (QUERY (a <* action |
                (SIZEOF (TYPEOF (a) *
                         ['AUTOMOTIVE_DESIGN.EXECUTED_ACTION',
                          'AUTOMOTIVE_DESIGN.EVENT_REFERENCE',
                          'AUTOMOTIVE_DESIGN.RENTENTION'])
                 > 1) OR
                ((SIZEOF (TYPEOF (a)) > 2) AND
                 (SIZEOF (TYPEOF (a) *
                         ['AUTOMOTIVE_DESIGN.EXECUTED_ACTION',
                          'AUTOMOTIVE_DESIGN.EVENT_REFERENCE',
                          'AUTOMOTIVE_DESIGN.RENTENTION'])
                  = 1))
        )) = 0;

END_RULE;
(*
```

Argument definitions:

action: the set of all instances of **action**.

Formal propositions:

WR1: Each instance of the subtypes of **action** shall be either one of **executed_action**, **event_reference**, **retention**, **property_process**, or **product_definition_process**, or shall be a complex instance of **property_process** and **product_definition_process**.

5.2.4.3 activity_requires_person_or_organization_assignment

The **activity_requires_person_or_organization_assignment** rule specifies that each instance of **action** which is not one of its subtypes and does not have a **description** of 'process operation' shall be referenced by at least one instance of either **applied_person_assignment** that has a **person_role** with a **name** of 'custodian', **applied_person_and_organization_assignment** that has a **person_and_organization_role** with a **name** of 'custodian', or **applied_organization_assignment** that has an **organization_role** with a **name** of 'custodian'. This rule enforces the requirement for every activity to have a person, an organization, or a person and an organization assigned to it that has the responsibility for the activity.

EXPRESS specification:

```
*)
RULE activity_requires_person_or_organization_assignment FOR
    (action);
WHERE
  WR1:  SIZEOF (QUERY (a <* action |
                (SIZEOF (TYPEOF (a)) = 1 ) AND
                (a.description <> 'process operation') AND
                (SIZEOF( QUERY( u <* USEDIN( a, 'AUTOMOTIVE_DESIGN.' +
                           'APPLIED_PERSON_ASSIGNMENT.ITEMS') |
                           (u.person_role.name = 'custodian')
                       )) +
                 SIZEOF( QUERY( u <* USEDIN( a, 'AUTOMOTIVE_DESIGN.' +
                           'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
                           (u.person_and_organization_role.name = 'custodian')
                       )) +
                 SIZEOF( QUERY( u <* USEDIN( a, 'AUTOMOTIVE_DESIGN.' +
                           'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
                           (u.organization_role.name = 'custodian')
                       ))
                 = 0 )
        )) = 0;

END_RULE;
(*
```

Argument definitions:

action: the set of all instances of **action**.

Formal propositions:

WR1: Each instance of **action** which is not one of its subtypes and does not have a **description** of 'process operation' shall be referenced by at least one instance of either **applied_person_assignment** that has a **person_role** with a **name** of 'custodian', **applied_person_and_organization_assignment** that has a **person_and_organization_role** with a **name** of 'custodian', or **applied_organization_assignment** that has an **organization_role** with a **name** of 'custodian'.

5.2.4.4 application_context_requires_ap_definition

The **application_context_requires_ap_definition** rule ensures that each instance of **application_context** shall be referenced by exactly one **application_protocol_definition.** The rule also specifies that the context for all product data in the scope of this part of ISO 10303 is associated with the schema name 'automotive_design'.

EXPRESS specification:

```
*)
RULE application_context_requires_ap_definition FOR
    (application_context);
WHERE
  WR1:   SIZEOF (QUERY (ac <* application_context |
         SIZEOF( QUERY( apd <* USEDIN( ac,'AUTOMOTIVE_DESIGN.' +
               'APPLICATION_PROTOCOL_DEFINITION.APPLICATION') |
         apd.application_interpreted_model_schema_name =
         'automotive_design'
         )) <> 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

application_context: the set of all instances of **application_context**.

Formal propositions:

WR1: For each instance of **application_context** there shall be exactly one instance of **application_protocol_definition** which contains the instance of **application_context** as its application attribute. Each **application_protocol_definition** shall specify the **application_interpreted_model_schema** with the **name** of 'automotive_design'.

5.2.4.5 approval_person_organization_requires_date_time

The **approval_person_organization_requires_date_time** rule specifies that each instance of **approval_person_organization** shall be referenced by exactly one of either **applied_date_assignment** or **applied_date_and_time_assignment** where the assigned date or date and time specifies the actual date of the approval. This rule enforces the requirement for every approval, that in case the approving person or organization is specified, also the actual date of the approval has to be given.

EXPRESS specification:

```
*)
RULE approval_person_organization_requires_date_time FOR
    (approval_person_organization);
WHERE
  WR1:  SIZEOF( QUERY( apo <* approval_person_organization |
         (SIZEOF( QUERY( u <* USEDIN( apo, 'AUTOMOTIVE_DESIGN.' +
            'APPLIED_DATE_ASSIGNMENT.ITEMS') |
            (u.date_role.name = 'actual')
                )) +
          SIZEOF( QUERY( u <* USEDIN( apo, 'AUTOMOTIVE_DESIGN.' +
            'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
            (u.date_time_role.name = 'actual')
                ))
         ) <> 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

approval_person_organization: the set of all instances of **approval_person_organization**.

Formal propositions:

WR1: There shall be exactly one instance of either **applied_date_assignment** or **applied_date_and_time_assignment** associated to an **approval_person_organization** where the **name** of **date_role/date_time_role** is 'actual'.

5.2.4.6 approval_requires_approval_assignment

The **approval_requires_approval_assignment** rule specifies that each instance of **approval** is referenced by at least one instance of **approval_assignment** as the **assigned_approval.** This rule enforces the requirement for every approval to be applied at least once.

EXPRESS specification:

```
*)
RULE approval_requires_approval_assignment FOR
    (approval);
WHERE
  WR1:  SIZEOF( QUERY( a <* approval |
                SIZEOF( USEDIN( a, 'AUTOMOTIVE_DESIGN.' +
                                   'APPROVAL_ASSIGNMENT.ASSIGNED_APPROVAL')
                      ) = 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

approval: the set of all instances of **approval**.

Formal propositions:

WR1: Each instance of **approval** shall be referenced at least once as the **assigned_approval** by an instance of **approval_assignment.**

5.2.4.7 characterized_object_subtype_exclusiveness

The **characterized_object_subtype_exclusiveness** rule specifies that an instance of the subtypes of **characterized_object** shall be one of **characterized_product_category**, **feature_component_definition**, **feature_definition**, **product_class**, or **product_component.**

EXPRESS specification:

```
*)
RULE characterized_object_subtype_exclusiveness FOR
    (characterized_object);
WHERE
  WR1:  SIZEOF (QUERY (co <* characterized_object |
                (SIZEOF (TYPEOF (co) *
                         ['AUTOMOTIVE_DESIGN.CHARACTERIZED_PRODUCT_CATEGORY',
                          'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION',
                          'AUTOMOTIVE_DESIGN.FEATURE_DEFINITION',
                          'AUTOMOTIVE_DESIGN.PRODUCT_CLASS',
                          'AUTOMOTIVE_DESIGN.PRODUCT_COMPONENT'])
                 > 1) OR
                ((SIZEOF (TYPEOF (co)) > 2) AND
                 (SIZEOF (TYPEOF (co) *
                          ['AUTOMOTIVE_DESIGN.CHARACTERIZED_PRODUCT_CATEGORY',
                           'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION',
                           'AUTOMOTIVE_DESIGN.FEATURE_DEFINITION',
```

```
                          'AUTOMOTIVE_DESIGN.PRODUCT_CLASS',
                          'AUTOMOTIVE_DESIGN.PRODUCT_COMPONENT'])
                  = 1 ))
        )) = 0;

END_RULE;
(*
```

Argument definitions:

characterized_object: the set of all instances of **characterized_object**.

Formal propositions:

WR1: Each instance of the subtypes of **characterized_object** shall be one of **characterized_product_category**, **feature_component_definition**, **feature_definition**, **product_class**, or **product_component.**

5.2.4.8 configuration_design_subtype_exclusiveness

The **configuration_design_subtype_exclusiveness** rule specifies that an instance of the subtypes of a **configuration_design** shall be only one of **component_function_definition**, **occurrence_usage_definition**, or **solution_alternative_definition**

EXPRESS specification:

```
*)
RULE configuration_design_subtype_exclusiveness FOR
    (configuration_design);
WHERE
  WR1:  SIZEOF (QUERY (pd <* configuration_design |
         (SIZEOF (TYPEOF (pd) *
         ['AUTOMOTIVE_DESIGN.component_function_definition',
          'AUTOMOTIVE_DESIGN.occurrence_usage_definition',
          'AUTOMOTIVE_DESIGN.solution_alternative_definition'])
                > 1) OR
         ((SIZEOF (TYPEOF (pd)) > 2) AND
         (SIZEOF (TYPEOF (pd) *
         ['AUTOMOTIVE_DESIGN.component_function_definition',
          'AUTOMOTIVE_DESIGN.occurrence_usage_definition',
          'AUTOMOTIVE_DESIGN.solution_alternative_definition'])
                 = 1))
        )) = 0;

END_RULE;
(*
```

Argument definitions:

configuration_design: the set of all instances of **configuration_design**.

Formal propositions:

WR1: Each instance of the subtypes of **configuration_design** shall be one of **component_function_definition**, **occurrence_usage_definition**, or **solution_alternative_definition**

5.2.4.9 coordinated_assembly_and_shape

The **coordinated_assembly_and_shape** rule specifies that the relationship between two **product_definition** entities which are an assembly and a component in a **next_assembly_usage_occurrence** and the relationship between two **shape_representation** entities which contain the representation of the shapes of the assembly and the component in a **shape_representation_relationship** must be explicitly related through the **context_dependent_shape_representation.** This rule calls the function **assembly_shape_is_defined** which returns true if the assembly relationship and shape relationship are explicitly related.

EXPRESS specification:

```
*)
RULE coordinated_assembly_and_shape FOR
    (next_assembly_usage_occurrence);
WHERE
  WR1:  SIZEOF (QUERY (nauo <* next_assembly_usage_occurrence |
        NOT (assembly_shape_is_defined (nauo)))) = 0;

END_RULE;
(*
```

Argument definitions:

next_assembly_usage_occurrence: the set of all instances of **next_assembly_usage_occurrence**.

Formal propositions:

WR1: For each instance of **next_assembly_usage_occurrence**, the **assembly_shape_is_defined** function shall be true.

5.2.4.10 curve_font_usage

The **curve_font_usage** rule ensures that each **curve_style_font**, **pre_defined_curve_font**, and **externally_defined_curve_font** shall define the curve font of at least one **curve_style.**

EXPRESS specification:

```
*)
RULE curve_font_usage FOR
    (curve_style_font,
     externally_defined_curve_font,
     pre_defined_curve_font);
WHERE
  WR1:  SIZEOF (QUERY (csf <* curve_style_font |
        NOT (SIZEOF (USEDIN (csf, 'AUTOMOTIVE_DESIGN.' +
                                  'CURVE_STYLE.CURVE_FONT'))>=1)))=0;
  WR2:  SIZEOF (QUERY (pdcf <* pre_defined_curve_font |
              NOT (SIZEOF (USEDIN (pdcf, 'AUTOMOTIVE_DESIGN.' +
                                         'CURVE_STYLE.CURVE_FONT'))>=1)))=0;
  WR3:  SIZEOF (QUERY (edcf <* externally_defined_curve_font |
              NOT (SIZEOF (USEDIN (edcf, 'AUTOMOTIVE_DESIGN.' +
                                         'CURVE_STYLE.CURVE_FONT'))>=1)))=0;

END_RULE;
(*
```

Argument definitions:

curve_style_font: the set of all instances of **curve_style_font**.

externally_defined_curve_font: the set of all instances of **externally_defined_curve_font**.

pre_defined_curve_font: the set of all instances of **pre_defined_curve_font**.

Formal propositions:

WR1: Each **curve_style_font** shall participate in the definition of at least one **curve_style.**

WR2: Each **pre_defined_curve_font** shall participate in the definition of at least one **curve_-style.**

WR3: Each **externally_defined_curve_font** shall participate in the definition of at least one **curve_style.**

5.2.4.11 definition_replacement_requires_effectivity_assignment

The **definition_replacement_requires_effectivity_assignment** rule specifies that each instance of **product_definition_relationship** with a **name** of 'definition replacement' shall be referenced by at least one instance of **applied_effective_item_assignment** that specifies the start of the **assigned_effectivity.** This rule enforces the requirement for every instance of **replaced_definition_relationship** to have at least one effectivity applied to it specifiying the start of the validity of the replacement.

EXPRESS specification:

```
*)
RULE definition_replacement_requires_effectivity_assignment FOR
    (product_definition_relationship);
WHERE
  WR1:  SIZEOF( QUERY( pdr <* product_definition_relationship |
                (pdr.name = 'definition replacement') AND
                (SIZEOF( QUERY(aeia <* USEDIN(pdr,'AUTOMOTIVE_DESIGN.' +
                                    'APPLIED_EFFECTIVE_ITEM_ASSIGNMENT.ITEMS') |
                         (('AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY' IN
                           TYPEOF(aeia.assigned_effectivity)) OR
                          ('AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY' IN
                           TYPEOF(aeia.assigned_effectivity))) AND
                         (SIZEOF( QUERY( ea <* USEDIN(aeia.assigned_effectivity,
                          'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT.ITEMS') |
                                  ea.action_role.name = 'effective start'
                         )) > 0)
                )) = 0)
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_definition_relationship: the set of all instances of **product_definition_relationship**.

Formal propositions:

WR1: For each instance of **product_definition_relationship** with a **name** of 'definition replacement', there shall be at least one instance of **applied_effective_item_assignment** that contains the instance in its set of **items**, and that has an **assigned_effectivity** that is a **dated_effectivity** or that has an **assigned_effectivity** that is an **event_based_effectivity** that is contained in the set of **items** of an **event_assignment** that references an **action_role** with a **name** of 'effectivity start'.

5.2.4.12 dependent_instantiable_action_request_role

The **dependent_instantiable_action_request_role** rule specifies that each instance of **action_request_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_action_request_role FOR
    (action_request_role);
```

```
WHERE
  WR1:  SIZEOF ( QUERY (a <* action_request_role |
        NOT (SIZEOF (USEDIN (a,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

action_request_role: identifies the set of all instances of **action_request_role.**

Formal propositions:

WR1: For each instance of **action_request_role,** there shall be a reference to the **action_request_role** instance from an attribute of another entity.

5.2.4.13 dependent_instantiable_action_role

The **dependent_instantiable_action_role** rule specifies that each instance of **action_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_action_role FOR
    (action_role);
WHERE
  WR1:  SIZEOF ( QUERY (a <* action_role |
        NOT (SIZEOF (USEDIN (a,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

action_role: identifies the set of all instances of **action_role.**

Formal propositions:

WR1: For each instance of **action_role,** there shall be a reference to the **action_role** instance from an attribute of another entity.

5.2.4.14 dependent_instantiable_approval_date_time_role

The **dependent_instantiable_approval_date_time_role** rule specifies that each instance of **approval_date_time_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_approval_date_time_role FOR
    (approval_date_time_role);
WHERE
  WR1:  SIZEOF ( QUERY (a <* approval_date_time_role |
        NOT (SIZEOF (USEDIN (a,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

approval_date_time_role: identifies the set of all instances of **approval_date_time_role.**

Formal propositions:

WR1: For each instance of **approval_date_time_role,** there shall be a reference to the **approval_date_time_role** instance from an attribute of another entity.

5.2.4.15 dependent_instantiable_approval_role

The **dependent_instantiable_approval_role** rule specifies that each instance of **approval_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_approval_role FOR
    (approval_role);
WHERE
  WR1:  SIZEOF ( QUERY (a <* approval_role |
        NOT (SIZEOF (USEDIN (a,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

approval_role: identifies the set of all instances of **approval_role.**

Formal propositions:

WR1: For each instance of **approval_role,** there shall be a reference to the **approval_role** instance from an attribute of another entity.

5.2.4.16 dependent_instantiable_approver_role

The **dependent_instantiable_approver_role** rule specifies that each instance of **approver_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_approver_role FOR
    (approver_role);
WHERE
  WR1:  SIZEOF ( QUERY (a <* approver_role |
        NOT (SIZEOF (USEDIN (a,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

approver_role: identifies the set of all instances of **approver_role.**

Formal propositions:

WR1: For each instance of **approver_role,** there shall be a reference to the **approver_role** instance from an attribute of another entity.

5.2.4.17 dependent_instantiable_certification_role

The **dependent_instantiable_certification_role** rule specifies that each instance of **certification_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_certification_role FOR
    (certification_role);
WHERE
  WR1:  SIZEOF ( QUERY (c <* certification_role |
        NOT (SIZEOF (USEDIN (c,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

certification_role: identifies the set of all instances of **certification_role.**

Formal propositions:

WR1: For each instance of **certification_role,** there shall be a reference to the **certification_-role** instance from an attribute of another entity.

5.2.4.18 dependent_instantiable_colour_rgb

The **dependent_instantiable_colour_rgb** rule specifies that each instance of **colour_rgb** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_colour_rgb FOR
    (colour_rgb);
WHERE
  WR1:  SIZEOF ( QUERY (c <* colour_rgb |
        NOT (SIZEOF (USEDIN (c,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

colour_rgb: identifies the set of all instances of **colour_rgb.**

Formal propositions:

WR1: For each instance of **colour_rgb,** there shall be a reference to the **colour_rgb** instance from an attribute of another entity.

5.2.4.19 dependent_instantiable_contract_role

The **dependent_instantiable_contract_role** rule specifies that each instance of **contract_-role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_contract_role FOR
    (contract_role);
WHERE
  WR1:  SIZEOF ( QUERY (c <* contract_role |
        NOT (SIZEOF (USEDIN (c,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

contract_role: identifies the set of all instances of **contract_role.**

Formal propositions:

WR1: For each instance of **contract_role,** there shall be a reference to the **contract_role** instance from an attribute of another entity.

5.2.4.20 dependent_instantiable_curve_style

The **dependent_instantiable_curve_style** rule specifies that each instance of **curve_style** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_curve_style FOR
    (curve_style);
WHERE
  WR1:  SIZEOF ( QUERY (c <* curve_style |
        NOT (SIZEOF (USEDIN (c,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

curve_style: identifies the set of all instances of **curve_style.**

Formal propositions:

WR1: For each instance of **curve_style,** there shall be a reference to the **curve_style** instance from an attribute of another entity.

5.2.4.21 dependent_instantiable_date

The **dependent_instantiable_date** rule specifies that each instance of **date** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_date FOR
    (date);
WHERE
```

```
  WR1:  SIZEOF ( QUERY (d <* date |
        NOT (SIZEOF (USEDIN (d,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

date: identifies the set of all instances of date.

Formal propositions:

WR1: For each instance of date, there shall be a reference to the date instance from an attribute of another entity.

5.2.4.22 dependent_instantiable_date_and_time

The **dependent_instantiable_date_and_time** rule specifies that each instance of **date_and_time** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_date_and_time FOR
    (date_and_time);
WHERE
  WR1:  SIZEOF ( QUERY (d <* date_and_time |
        NOT (SIZEOF (USEDIN (d,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

date_and_time: identifies the set of all instances of **date_and_time.**

Formal propositions:

WR1: For each instance of **date_and_time,** there shall be a reference to the **date_and_time** instance from an attribute of another entity.

5.2.4.23 dependent_instantiable_date_role

The **dependent_instantiable_date_role** rule specifies that each instance of **date_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_date_role FOR
    (date_role);
WHERE
  WR1:  SIZEOF ( QUERY (d <* date_role |
        NOT (SIZEOF (USEDIN (d,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

date_role: identifies the set of all instances of **date_role.**

Formal propositions:

WR1: For each instance of **date_role,** there shall be a reference to the **date_role** instance from an attribute of another entity.

5.2.4.24 dependent_instantiable_date_time_role

The **dependent_instantiable_date_time_role** rule specifies that each instance of **date_time_-role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_date_time_role FOR
    (date_time_role);
WHERE
  WR1:  SIZEOF ( QUERY (d <* date_time_role |
        NOT (SIZEOF (USEDIN (d,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

date_time_role: identifies the set of all instances of **date_time_role.**

Formal propositions:

WR1: For each instance of **date_time_role,** there shall be a reference to the **date_time_role** instance from an attribute of another entity.

5.2.4.25 dependent_instantiable_derived_unit

The **dependent_instantiable_derived_unit** rule specifies that each instance of **derived_unit** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_derived_unit FOR
    (derived_unit);
WHERE
  WR1:  SIZEOF ( QUERY (d <* derived_unit |
        NOT (SIZEOF (USEDIN (d,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

derived_unit: identifies the set of all instances of **derived_unit.**

Formal propositions:

WR1: For each instance of **derived_unit,** there shall be a reference to the **derived_unit** instance from an attribute of another entity.

5.2.4.26 dependent_instantiable_dimensional_size

The **dependent_instantiable_dimensional_size** rule specifies that each instance of **dimensional_size** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_dimensional_size FOR
    (dimensional_size);
WHERE
  WR1:  SIZEOF ( QUERY (d <* dimensional_size |
        NOT (SIZEOF (USEDIN (d,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

dimensional_size: identifies the set of all instances of **dimensional_size.**

Formal propositions:

WR1: For each instance of **dimensional_size,** there shall be a reference to the **dimensional_size** instance from an attribute of another entity.

5.2.4.27 dependent_instantiable_document_role

The **dependent_instantiable_document_role** rule specifies that each instance of **document_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_document_role FOR
    (document_role);
WHERE
  WR1:  SIZEOF ( QUERY (d <* document_role |
        NOT (SIZEOF (USEDIN (d,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

document_role: identifies the set of all instances of **document_role.**

Formal propositions:

WR1: For each instance of **document_role,** there shall be a reference to the **document_role** instance from an attribute of another entity.

5.2.4.28 dependent_instantiable_effectivity_role

The **dependent_instantiable_effectivity_role** rule specifies that each instance of **effectivity_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_effectivity_role FOR
    (effectivity_role);
WHERE
  WR1:  SIZEOF ( QUERY (e <* effectivity_role |
        NOT (SIZEOF (USEDIN (e,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

effectivity_role: identifies the set of all instances of **effectivity_role.**

Formal propositions:

WR1: For each instance of **effectivity_role,** there shall be a reference to the **effectivity_role** instance from an attribute of another entity.

5.2.4.29 dependent_instantiable_externally_defined_symbol

The **dependent_instantiable_externally_defined_symbol** rule specifies that each instance of **externally_defined_symbol** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_externally_defined_symbol FOR
    (externally_defined_symbol);
WHERE
  WR1:  SIZEOF ( QUERY (e <* externally_defined_symbol |
        NOT (SIZEOF (USEDIN (e,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

externally_defined_symbol: identifies the set of all instances of **externally_defined_symbol.**

Formal propositions:

WR1: For each instance of **externally_defined_symbol,** there shall be a reference to the **externally_defined_symbol** instance from an attribute of another entity.

5.2.4.30 dependent_instantiable_fill_area_style

The **dependent_instantiable_fill_area_style** rule specifies that each instance of **fill_area_-style** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_fill_area_style FOR
    (fill_area_style);
WHERE
```

```
  WR1:  SIZEOF ( QUERY (f <* fill_area_style |
        NOT (SIZEOF (USEDIN (f,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

fill_area_style: identifies the set of all instances of **fill_area_style.**

Formal propositions:

WR1: For each instance of **fill_area_style,** there shall be a reference to the **fill_area_style** instance from an attribute of another entity.

5.2.4.31 dependent_instantiable_fill_area_style_colour

The **dependent_instantiable_fill_area_style_colour** rule specifies that each instance of **fill_area_style_colour** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_fill_area_style_colour FOR
    (fill_area_style_colour);
WHERE
  WR1:  SIZEOF ( QUERY (f <* fill_area_style_colour |
        NOT (SIZEOF (USEDIN (f,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

fill_area_style_colour: identifies the set of all instances of **fill_area_style_colour.**

Formal propositions:

WR1: For each instance of **fill_area_style_colour,** there shall be a reference to the **fill_area_style_colour** instance from an attribute of another entity.

5.2.4.32 dependent_instantiable_group_role

The **dependent_instantiable_group_role** rule specifies that each instance of **group_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_group_role FOR
    (group_role);
WHERE
  WR1:  SIZEOF ( QUERY (g <* group_role |
        NOT (SIZEOF (USEDIN (g,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

group_role: identifies the set of all instances of **group_role.**

Formal propositions:

WR1: For each instance of **group_role,** there shall be a reference to the **group_role** instance from an attribute of another entity.

5.2.4.33 dependent_instantiable_identification_role

The **dependent_instantiable_identification_role** rule specifies that each instance of **identification_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_identification_role FOR
    (identification_role);
WHERE
  WR1:  SIZEOF ( QUERY (i <* identification_role |
        NOT (SIZEOF (USEDIN (i,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

identification_role: identifies the set of all instances of **identification_role.**

Formal propositions:

WR1: For each instance of **identification_role,** there shall be a reference to the **identification_role** instance from an attribute of another entity.

5.2.4.34 dependent_instantiable_language_role

The **dependent_instantiable_language_role** rule specifies that each instance of **language_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_language_role FOR
    (language_role);
WHERE
  WR1:  SIZEOF ( QUERY (l <* language_role |
        NOT (SIZEOF (USEDIN (l,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

language_role: identifies the set of all instances of **language_role.**

Formal propositions:

WR1: For each instance of **language_role,** there shall be a reference to the **language_role** instance from an attribute of another entity.

5.2.4.35 dependent_instantiable_measure_with_unit

The **dependent_instantiable_measure_with_unit** rule specifies that each instance of **measure_with_unit** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_measure_with_unit FOR
    (measure_with_unit);
WHERE
  WR1:  SIZEOF ( QUERY (m <* measure_with_unit |
        NOT (SIZEOF (USEDIN (m,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

measure_with_unit: identifies the set of all instances of **measure_with_unit.**

Formal propositions:

WR1: For each instance of **measure_with_unit**, there shall be a reference to the **measure_with_unit** instance from an attribute of another entity.

5.2.4.36 dependent_instantiable_named_unit

The **dependent_instantiable_named_unit** rule specifies that each instance of **named_unit** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_named_unit FOR
    (named_unit);
WHERE
  WR1:  SIZEOF ( QUERY (n <* named_unit |
        NOT (SIZEOF (USEDIN (n,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

named_unit: identifies the set of all instances of **named_unit.**

Formal propositions:

WR1: For each instance of **named_unit**, there shall be a reference to the **named_unit** instance from an attribute of another entity.

5.2.4.37 dependent_instantiable_organization_role

The **dependent_instantiable_organization_role** rule specifies that each instance of **organization_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_organization_role FOR
    (organization_role);
WHERE
  WR1:  SIZEOF ( QUERY (o <* organization_role |
        NOT (SIZEOF (USEDIN (o,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

organization_role: identifies the set of all instances of **organization_role.**

Formal propositions:

WR1: For each instance of **organization_role,** there shall be a reference to the **organization_role** instance from an attribute of another entity.

5.2.4.38 dependent_instantiable_person_and_organization_role

The **dependent_instantiable_person_and_organization_role** rule specifies that each instance of **person_and_organization_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_person_and_organization_role FOR
    (person_and_organization_role);
WHERE
  WR1:  SIZEOF ( QUERY (p <* person_and_organization_role |
        NOT (SIZEOF (USEDIN (p,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

person_and_organization_role: identifies the set of all instances of **person_and_organization_role.**

Formal propositions:

WR1: For each instance of **person_and_organization_role,** there shall be a reference to the **person_and_organization_role** instance from an attribute of another entity.

5.2.4.39 dependent_instantiable_person_role

The **dependent_instantiable_person_role** rule specifies that each instance of **person_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_person_role FOR
    (person_role);
```

```
WHERE
  WR1:  SIZEOF ( QUERY (p <* person_role |
        NOT (SIZEOF (USEDIN (p,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

person_role: identifies the set of all instances of **person_role.**

Formal propositions:

WR1: For each instance of **person_role,** there shall be a reference to the **person_role** instance from an attribute of another entity.

5.2.4.40 dependent_instantiable_pre_defined_colour

The **dependent_instantiable_pre_defined_colour** rule specifies that each instance of **pre_defined_colour** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_pre_defined_colour FOR
    (pre_defined_colour);
WHERE
  WR1:  SIZEOF ( QUERY (p <* pre_defined_colour |
        NOT (SIZEOF (USEDIN (p,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

pre_defined_colour: identifies the set of all instances of **pre_defined_colour.**

Formal propositions:

WR1: For each instance of **pre_defined_colour,** there shall be a reference to the **pre_defined_colour** instance from an attribute of another entity.

5.2.4.41 dependent_instantiable_pre_defined_symbol

The **dependent_instantiable_pre_defined_symbol** rule specifies that each instance of **pre_defined_symbol** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_pre_defined_symbol FOR
    (pre_defined_symbol);
WHERE
  WR1:  SIZEOF ( QUERY (p <* pre_defined_symbol |
        NOT (SIZEOF (USEDIN (p,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

pre_defined_symbol: identifies the set of all instances of **pre_defined_symbol.**

Formal propositions:

WR1: For each instance of **pre_defined_symbol,** there shall be a reference to the **pre_defined_symbol** instance from an attribute of another entity.

5.2.4.42 dependent_instantiable_presentation_style_by_context

The **dependent_instantiable_presentation_style_by_context** rule specifies that each instance of **presentation_style_by_context** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_presentation_style_by_context FOR
    (presentation_style_by_context);
WHERE
  WR1:  SIZEOF ( QUERY (p <* presentation_style_by_context |
        NOT (SIZEOF (USEDIN (p,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

presentation_style_by_context: identifies the set of all instances of **presentation_style_by_context.**

Formal propositions:

WR1: For each instance of **presentation_style_by_context,** there shall be a reference to the **presentation_style_by_context** instance from an attribute of another entity.

5.2.4.43 dependent_instantiable_representation

The **dependent_instantiable_representation** rule specifies that each instance of **representation** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_representation FOR
    (representation);
WHERE
  WR1:  SIZEOF ( QUERY (r <* representation |
        NOT (SIZEOF (USEDIN (r,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

representation: identifies the set of all instances of representation.

Formal propositions:

WR1: For each instance of representation, there shall be a reference to the representation instance from an attribute of another entity.

5.2.4.44 dependent_instantiable_security_classification_role

The **dependent_instantiable_security_classification_role** rule specifies that each instance of **security_classification_role** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_security_classification_role FOR
    (security_classification_role);
WHERE
  WR1:  SIZEOF ( QUERY (s <* security_classification_role |
        NOT (SIZEOF (USEDIN (s,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

security_classification_role: identifies the set of all instances of **security_classification_role.**

Formal propositions:

WR1: For each instance of **security_classification_role**, there shall be a reference to the **security_classification_role** instance from an attribute of another entity.

5.2.4.45 dependent_instantiable_symbol_colour

The **dependent_instantiable_symbol_colour** rule specifies that each instance of **symbol_colour** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_symbol_colour FOR
    (symbol_colour);
WHERE
  WR1:  SIZEOF ( QUERY (s <* symbol_colour |
        NOT (SIZEOF (USEDIN (s,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

symbol_colour: identifies the set of all instances of **symbol_colour.**

Formal propositions:

WR1: For each instance of **symbol_colour**, there shall be a reference to the **symbol_colour** instance from an attribute of another entity.

5.2.4.46 dependent_instantiable_text_style

The **dependent_instantiable_text_style** rule specifies that each instance of **text_style** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_text_style FOR
    (text_style);
WHERE
  WR1:  SIZEOF ( QUERY (t <* text_style |
        NOT (SIZEOF (USEDIN (t,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

text_style: identifies the set of all instances of **text_style.**

Formal propositions:

WR1: For each instance of **text_style,** there shall be a reference to the **text_style** instance from an attribute of another entity.

5.2.4.47 dependent_instantiable_text_style_for_defined_font

The **dependent_instantiable_text_style_for_defined_font** rule specifies that each instance of **text_style_for_defined_font** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_text_style_for_defined_font FOR
    (text_style_for_defined_font);
WHERE
  WR1:  SIZEOF ( QUERY (t <* text_style_for_defined_font |
        NOT (SIZEOF (USEDIN (t,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

text_style_for_defined_font: identifies the set of all instances of **text_style_for_defined_-font.**

Formal propositions:

WR1: For each instance of **text_style_for_defined_font,** there shall be a reference to the **text_style_for_defined_font** instance from an attribute of another entity.

5.2.4.48 dependent_instantiable_tolerance_value

The **dependent_instantiable_tolerance_value** rule specifies that each instance of **tolerance_value** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_tolerance_value FOR
    (tolerance_value);
```

```
WHERE
  WR1:  SIZEOF ( QUERY (t <* tolerance_value |
        NOT (SIZEOF (USEDIN (t,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

tolerance_value: identifies the set of all instances of **tolerance_value.**

Formal propositions:

WR1: For each instance of **tolerance_value,** there shall be a reference to the **tolerance_value** instance from an attribute of another entity.

5.2.4.49 dependent_instantiable_uncertainty_qualifier

The **dependent_instantiable_uncertainty_qualifier** rule specifies that each instance of **uncertainty_qualifier** is dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_uncertainty_qualifier FOR
    (uncertainty_qualifier);
WHERE
  WR1:  SIZEOF ( QUERY (u <* uncertainty_qualifier |
        NOT (SIZEOF (USEDIN (u,'')) > 0 ))) = 0;

END_RULE;
(*
```

Argument definitions:

uncertainty_qualifier: identifies the set of all instances of **uncertainty_qualifier.**

Formal propositions:

WR1: For each instance of **uncertainty_qualifier,** there shall be a reference to the **uncertainty_qualifier** instance from an attribute of another entity.

5.2.4.50 dimensionality_is_two_or_three

The **dimensionality_is_two_or_three** rule ensures that all **representations** with a **geometric_representation_context** are defined in a two-dimensional or three-dimensional coordinate

system.

EXPRESS specification:

```
*)
RULE dimensionality_is_two_or_three FOR
    (geometric_representation_context);
WHERE
  WR1:  SIZEOF (QUERY (g <* geometric_representation_context |
                 NOT ((g.coordinate_space_dimension = 2) OR
                      (g.coordinate_space_dimension = 3))
               )) = 0;

END_RULE;
(*
```

Argument definitions:

geometric_representation_context: the set of all instances of **geometric_representation_-context**.

Formal propositions:

WR1: Each **geometric_representation_context** shall have a **coordinate_space_dimension** equal to two or three.

5.2.4.51 draughting_model_annotation_layers

The **draughting_model_annotation_layers** rule ensures that every **annotation_occurrence** that is an item in a **draughting_model** is assigned to at least one layer.

EXPRESS specification:

```
*)
RULE draughting_model_annotation_layers FOR
    (draughting_model);
WHERE
  WR1:  SIZEOF
               (QUERY (dm <* draughting_model |
               NOT (SIZEOF (QUERY (ao <* QUERY (it <* dm\representation.items |
                  ('AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE' IN TYPEOF (it))) |
                  NOT (SIZEOF (USEDIN (ao, 'AUTOMOTIVE_DESIGN.' +
                                           'PRESENTATION_LAYER_ASSIGNMENT.' +
                                           'ASSIGNED_ITEMS')) >= 1)
                   ))=0)
               ))=0;

END_RULE;
```

```
(*
```

Argument definitions:

draughting_model: the set of all instances of **draughting_model**.

Formal propositions:

WR1: Every item of each **draughting_model** that is an **annotation_occurrence** shall be one of the **assigned_items** of at least one **presentation_layer_assignment.**

5.2.4.52 draughting_subfigure_representation_layers

The **draughting_subfigure_representation_layers** rule ensures that every **annotation_occurrence** that is an item in a **draughting_subfigure_representation** is assigned to at least one layer.

EXPRESS specification:

```
*)
RULE draughting_subfigure_representation_layers FOR
    (draughting_subfigure_representation);
WHERE
  WR1:  SIZEOF (QUERY (dsr <* draughting_subfigure_representation |
               NOT (SIZEOF (QUERY (item <* dsr\representation.items |
                   ('AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE'
                    IN TYPEOF (item)) AND
                   (SIZEOF (USEDIN (item, 'AUTOMOTIVE_DESIGN.' +
                                          'PRESENTATION_LAYER_ASSIGNMENT.' +
                                          'ASSIGNED_ITEMS'))=0)
                   ))=0)
               ))=0;

END_RULE;
(*
```

Argument definitions:

draughting_subfigure_representation: the set of all instances of **draughting_subfigure_representation**.

Formal propositions:

WR1: Every item of each **draughting_subfigure_representation** that is an **annotation_occurrence** shall be one of the **assigned_items** of at least one **presentation_layer_assignment.**

5.2.4.53 drawing_sheet_annotation_layers

The **drawing_sheet_annotation_layers** rule ensures that every **annotation_occurrence** that is an item in a **drawing_sheet_revision** is assigned to at least one layer.

EXPRESS specification:

```
*)
RULE drawing_sheet_annotation_layers FOR
    (drawing_sheet_revision);
WHERE
  WR1:  SIZEOF
               (QUERY (dsr <* drawing_sheet_revision |
               NOT (SIZEOF (QUERY (ao <* QUERY (it <* dsr\representation.items |
                  ('AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE' IN TYPEOF (it))) |
                  NOT (SIZEOF (USEDIN (ao, 'AUTOMOTIVE_DESIGN.' +
                                           'PRESENTATION_LAYER_ASSIGNMENT.' +
                                           'ASSIGNED_ITEMS')) >= 1)
                   ))=0)
               ))=0;

END_RULE;
(*
```

Argument definitions:

drawing_sheet_revision: the set of all instances of **drawing_sheet_revision**.

Formal propositions:

WR1: Every item of each **drawing_sheet_revision** that is an **annotation_occurrence** shall be one of the **assigned_items** of at least one **presentation_layer_assignment.**

5.2.4.54 drawing_sheet_layout_usage

The **drawing_sheet_layout_usage** rule ensures that a **drawing_sheet_layout** is mapped as one of the **items** of a **drawing_sheet_revision.**

EXPRESS specification:

```
*)
RULE drawing_sheet_layout_usage FOR
    (mapped_item);
WHERE
  WR1:  SIZEOF (QUERY (dsl <* QUERY (mi <* mapped_item |
                 ('AUTOMOTIVE_DESIGN.DRAWING_SHEET_LAYOUT'
                     IN TYPEOF (mi.mapping_source.mapped_representation))) |
                 NOT (SIZEOF (USEDIN (dsl, '')) =
```

```
                    SIZEOF (QUERY (dsr <* USEDIN (dsl, 'AUTOMOTIVE_DESIGN.' +
                                        'REPRESENTATION.ITEMS') |
                    ('AUTOMOTIVE_DESIGN.DRAWING_SHEET_REVISION'
                       IN TYPEOF(dsr)))))
               )) = 0;

END_RULE;
(*
```

Argument definitions:

mapped_item: the set of all instances of **mapped_item**.

Formal propositions:

WR1: Each **mapped_item** that is the mapping of a **drawing_sheet_layout** shall be one of the items in a **drawing_sheet_revision.**

5.2.4.55 drawing_view_annotation_layers

The **drawing_view_annotation_layers** rule ensures that every **annotation_occurrence** that is an item in a **presentation_view** is assigned to at least one layer.

EXPRESS specification:

```
*)
RULE drawing_view_annotation_layers FOR
    (presentation_view);
WHERE
  WR1:  SIZEOF
               (QUERY (pv <* presentation_view |
               NOT (SIZEOF (QUERY (ao <* QUERY (it <* pv\representation.items |
                  ('AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE' IN TYPEOF (it))) |
                  NOT (SIZEOF (USEDIN (ao, 'AUTOMOTIVE_DESIGN.' +
                                          'PRESENTATION_LAYER_ASSIGNMENT.' +
                                          'ASSIGNED_ITEMS')) >= 1)
                   ))=0)
               ))=0;

END_RULE;
(*
```

Argument definitions:

presentation_view: the set of all instances of **presentation_view**.

Formal propositions:

WR1: Every item of each **presentation_view** that is an **annotation_occurrence** shall be one of the **assigned_items** of at least one **presentation_layer_assignment.**

5.2.4.56 effective_context_assignment_subtype_exclusiveness

The **effective_context_assignment_subtype_exclusiveness** rule specifies that an instance of the subtypes of **effective_context_assignment** shall be one of **applied_effective_context_-assignment**, or **configured_effective_context_assignment.**

EXPRESS specification:

```
*)
RULE effective_context_assignment_subtype_exclusiveness FOR
    (effective_context_assignment);
WHERE
  WR1:  SIZEOF (QUERY (eca <* effective_context_assignment |
                (SIZEOF (TYPEOF (eca) *
                         ['AUTOMOTIVE_DESIGN.APPLIED_EFFECTIVE_CONTEXT_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.CONFIGURED_EFFECTIVE_CONTEXT_ASSIGNMENT'])
                 > 1) OR
                ((SIZEOF (TYPEOF (eca)) > 2) AND
                 (SIZEOF (TYPEOF (eca) *
                          ['AUTOMOTIVE_DESIGN.APPLIED_EFFECTIVE_CONTEXT_ASSIGNMENT',
                           'AUTOMOTIVE_DESIGN.CONFIGURED_EFFECTIVE_CONTEXT_ASSIGNMENT'])
                  = 1 ))
        )) = 0;

END_RULE;
(*
```

Argument definitions:

effective_context_assignment: the set of all instances of **effective_context_assignment**.

Formal propositions:

WR1: Each instance of the subtypes of **effective_context_assignment** shall be one of **applied_effective_context_assignment**, or **configured_effective_context_assignment.**

5.2.4.57 effective_item_assignment_subtype_exclusiveness

The **effective_item_assignment_subtype_exclusiveness** rule specifies that an instance of the subtypes of **effective_item_assignment** shall be one of **applied_effective_item_assignment**, or **configured_effective_item_assignment.**

EXPRESS specification:

```
*)
RULE effective_item_assignment_subtype_exclusiveness FOR
    (effective_item_assignment);
WHERE
  WR1:  SIZEOF (QUERY (eia <* effective_item_assignment |
                (SIZEOF (TYPEOF (eia) *
                         ['AUTOMOTIVE_DESIGN.APPLIED_EFFECTICE_ITEM_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.CONFIGURED_EFFECTIVE_ITEM_ASSIGNMENT'])
                 > 1) OR
                ((SIZEOF (TYPEOF (eia)) > 2) AND
                 (SIZEOF (TYPEOF (eia) *
                          ['AUTOMOTIVE_DESIGN.APPLIED_EFFECTICE_ITEM_ASSIGNMENT',
                           'AUTOMOTIVE_DESIGN.CONFIGURED_EFFECTIVE_ITEM_ASSIGNMENT'])
                  = 1 ))
        )) = 0;

END_RULE;
(*
```

Argument definitions:

effective_item_assignment: the set of all instances of **effective_item_assignment**.

Formal propositions:

WR1: Each instance of the subtypes of **effective_item_assignment** shall be one of **applied_-effective_item_assignment**, or **configured_effective_item_assignment.**

5.2.4.58 effectivity_requires_effective_item_assignment

The **effectivity_requires_effective_item_assignment** rule specifies that each instance of effectivity is referenced by at least one instance of **effective_item_assignment** as the **assigned_-effectivity.** This rule enforces the requirement for every effectivity to be applied at least once and for every configuration to specify the usage of a configured item.

EXPRESS specification:

```
*)
RULE effectivity_requires_effective_item_assignment FOR
    (effectivity);
WHERE
  WR1:  SIZEOF( QUERY( e <* effectivity |
                SIZEOF( USEDIN( e, 'AUTOMOTIVE_DESIGN.' +
                        'EFFECTIVE_ITEM_ASSIGNMENT.ASSIGNED_EFFECTIVITY')
                ) < 1
        )) = 0;
```

```
END_RULE;
(*
```

Argument definitions:

effectivity: the set of all instances of **effectivity**.

Formal propositions:

WR1: Each instance of effectivity shall be used at least once as the **assigned_effectivity** by an instance of **effective_item_assignment.**

5.2.4.59 fill_area_style_tile_symbol_constraint

The **fill_area_style_tile_symbol_constraint** rule ensures that a **pre_defined_symbol** does not participate in the definition of a **fill_area_style_tile_symbol_with_style.**

EXPRESS specification:

```
*)
RULE fill_area_style_tile_symbol_constraint FOR
    (fill_area_style_tile_symbol_with_style);
WHERE
  WR1:  SIZEOF (QUERY (fast <* fill_area_style_tile_symbol_with_style |
                 (('AUTOMOTIVE_DESIGN.DEFINED_SYMBOL' IN
                     TYPEOF(fast.symbol\styled_item.item)) AND
                 NOT ('AUTOMOTIVE_DESIGN.EXTERNALLY_DEFINED_SYMBOL' IN
                         TYPEOF(fast.symbol\styled_item.
                                item\defined_symbol.definition)))
               )) = 0;

END_RULE;
(*
```

Argument definitions:

fill_area_style_tile_symbol_with_style: the set of all instances of **fill_area_style_tile_symbol_-with_style**.

Formal propositions:

WR1: Each **fill_area_style_tile_symbol_with_style** that has an implicit definition shall be defined by an **externally_defined_symbol.**

5.2.4.60 functional_product_definition_requires_functional_product

The **functional_product_definition_requires_functional_product** rule specifies that each instance of **product_definition** that has as its **frame_of_reference** a **product_definition_context** with a **name** of 'functional definition' shall be associated to a **product_definition_formation** that references a product that has a **product_context** with a **name** of 'functionality' as frame of reference.

EXPRESS specification:

```
*)
RULE functional_product_definition_requires_functional_product FOR
    (product_definition);
WHERE
  WR1:  SIZEOF( QUERY( pd <* product_definition |
                (pd.frame_of_reference.name = 'functional definition') AND
                (SIZEOF( QUERY( pc <* pd.formation.of_product.frame_of_reference |
                         pc.name = 'functionality'
                 )) < 1 )
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_definition: the set of all instances of **product_definition**.

Formal propositions:

WR1: Each instance of **product_definition** that has as its **frame_of_reference** a **product_definition_context** with a **name** of 'functional definition' shall be associated to a **product_definition_formation** that references a product that has a **product_context** with a **name** of 'functionality' in its set of **product_context** instances.

5.2.4.61 geometric_tolerance_subtype_exclusiveness

The **geometric_tolerance_subtype_exclusiveness** rule specifies that an instance of the subtypes of a **geometric_tolerance** shall be only one of **angularity_tolerance**, **circular_runout_tolerance**, **circularity_tolerance**, **concentricity_tolerance**, **cylindricity_tolerance**, **flatness_tolerance**, **linear_profile_tolerance**, **parallelism_tolerance**, **perpendicularity_tolerance**, **position_tolerance**, **straightness_tolerance**, **surface_profile_tolerance**, **symmetry_tolerance**, or **total_runout_tolerance**.

EXPRESS specification:

```
*)
RULE geometric_tolerance_subtype_exclusiveness FOR
    (geometric_tolerance);
WHERE
  WR1:  SIZEOF (QUERY (pd <* geometric_tolerance |
                (SIZEOF (TYPEOF (pd) *
                         ['AUTOMOTIVE_DESIGN.ANGULARITY_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.CIRCULAR_RUNOUT_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.CIRCULARITY_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.CONCENTRICITY_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.CYLINDRICITY_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.FLATNESS_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.LINEAR_PROFILE_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.PARALLELISM_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.PERPENDICULARITY_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.POSITION_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.STRAIGHTNESS_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.SURFACE_PROFILE_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.SYMMETRY_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.TOTAL_RUNOUT_TOLERANCE'])
                 > 1) OR
                ((SIZEOF (TYPEOF (pd)) > 2) AND
                 (SIZEOF (TYPEOF (pd) *
                         ['AUTOMOTIVE_DESIGN.ANGULARITY_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.CIRCULAR_RUNOUT_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.CIRCULARITY_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.CONCENTRICITY_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.CYLINDRICITY_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.FLATNESS_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.LINEAR_PROFILE_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.PARALLELISM_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.PERPENDICULARITY_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.POSITION_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.STRAIGHTNESS_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.SURFACE_PROFILE_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.SYMMETRY_TOLERANCE',
                          'AUTOMOTIVE_DESIGN.TOTAL_RUNOUT_TOLERANCE'])
                  = 1))
        )) = 0;


END_RULE;
(*
```

Argument definitions:

geometric_tolerance: the set of all instances of **geometric_tolerance**.

Formal propositions:

WR1: Each instance of the subtypes of **geometric_tolerance** shall be one of **angularity_tolerance**, **circular_runout_tolerance**, **circularity_tolerance**, **concentricity_tolerance**, **cylindricity_tolerance**, **flatness_tolerance**, **linear_profile_tolerance**, **parallelism_tolerance**, **perpendicularity_tolerance**, **position_tolerance**, **straightness_tolerance**, **surface_profile_tolerance**, **symmetry_tolerance**, or **total_runout_tolerance.**

5.2.4.62 global_length_and_angle_units_2d_or_3d

The **global_length_and_angle_units_2d_or_3d** rule ensures that units of length and plane angle are given for every **global_unit_assigned_context.**

EXPRESS specification:

```
*)
RULE global_length_and_angle_units_2d_or_3d FOR
    (global_unit_assigned_context);
WHERE
  WR1:  SIZEOF (QUERY (guac <* global_unit_assigned_context |
                 NOT (SIZEOF (guac.units) >= 2)
              ))=0;
  WR2:   SIZEOF (QUERY (guac <* global_unit_assigned_context |
              NOT ((SIZEOF (QUERY (x <* guac.units |
                    'AUTOMOTIVE_DESIGN.LENGTH_UNIT' IN TYPEOF (x)))=1) AND
                   (SIZEOF (QUERY (x <* guac.units |
                    'AUTOMOTIVE_DESIGN.PLANE_ANGLE_UNIT' IN TYPEOF (x)))=1)
              )))=0;

END_RULE;
(*
```

Argument definitions:

global_unit_assigned_context: the set of all instances of **global_unit_assigned_context**.

Formal propositions:

WR1: Each **global_unit_assigned_context** shall have at least two elements in its set of **units**.

WR2: Each **global_unit_assigned_context** shall have exactly one **length_unitand** exactly one **plane_angle_unit.**

5.2.4.63 group_assignment_subtype_exclusiveness

The **group_assignment_subtype_exclusiveness** rule specifies that an instance of the subtypes of **group_assignment** shall be one of **applied_group_assignment**, **group_member_-**

assignment, or **product_concept_feature_category_usage.**

EXPRESS specification:

```
*)
RULE group_assignment_subtype_exclusiveness FOR
    (group_assignment);
WHERE
  WR1:  SIZEOF (QUERY (ga <* group_assignment |
                (SIZEOF (TYPEOF (ga) *
                         ['AUTOMOTIVE_DESIGN.APPLIED_GROUP_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.GROUP_MEMBER_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_FEATURE_CATEGORY_USAGE'])
                 > 1) OR
                ((SIZEOF (TYPEOF (ga)) > 2) AND
                 (SIZEOF (TYPEOF (ga) *
                         ['AUTOMOTIVE_DESIGN.APPLIED_GROUP_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.GROUP_MEMBER_ASSIGNMENT',
                          'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_FEATURE_CATEGORY_USAGE'])
                  = 1))
        )) = 0;

END_RULE;
(*
```

Argument definitions:

group_assignment: the set of all instances of **group_assignment**.

Formal propositions:

WR1: Each instance of the subtypes of **group_assignment** shall be one of **applied_group_-assignment**, **group_member_assignment**, or **product_concept_feature_category_usage.**

5.2.4.64 identification_assignment_subtype_exclusiveness

The **identification_assignment_subtype_exclusiveness** rule specifies that an instance of the subtypes of **identification_assignment** shall be one of **applied_identification_assignment**, **applied_external_identification_assignment**, or **version_assignment.**

EXPRESS specification:

```
*)
RULE identification_assignment_subtype_exclusiveness FOR
    (identification_assignment);
WHERE
  WR1:  SIZEOF (QUERY (ia <* identification_assignment |
                (SIZEOF (TYPEOF (ia) *
```

```
                        ['AUTOMOTIVE_DESIGN.APPLIED_IDENTIFICATION_ASSIGNMENT',
                         'AUTOMOTIVE_DESIGN.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT',
                         'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT'])
                 > 1) OR
                ((SIZEOF (TYPEOF (ia)) > 2) AND
                 (SIZEOF (TYPEOF (ia) *
                        ['AUTOMOTIVE_DESIGN.APPLIED_IDENTIFICATION_ASSIGNMENT',
                         'AUTOMOTIVE_DESIGN.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT',
                         'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT'])
                  = 1))
        )) = 0;

END_RULE;
(*
```

Argument definitions:

identification_assignment: the set of all instances of **identification_assignment**.

Formal propositions:

WR1: Each instance of the subtypes of **identification_assignment** shall be one of **applied_identification_assignment**, **applied_external_identification_assignment**, or **version_assignment.**

5.2.4.65 organization_assignment_subtype_exclusiveness

The **organization_assignment_subtype_exclusiveness** rule specifies that an instance of the subtypes of **organization_assignment** shall be one of **applied_date_and_organization_assignment**, **applied_organization_assignment**, or **project_assignment.**

EXPRESS specification:

```
*)
RULE organization_assignment_subtype_exclusiveness FOR
    (organization_assignment);
WHERE
  WR1:  SIZEOF (QUERY (oa <* organization_assignment |
                (SIZEOF (TYPEOF (oa) *
                        ['AUTOMOTIVE_DESIGN.APPLIED_DATE_AND_ORGANIZATION_ASSIGNMENT',
                         'AUTOMOTIVE_DESIGN.APPLIED_ORGANIZATION_ASSIGNMENT',
                         'AUTOMOTIVE_DESIGN.PROJECT_ASSIGNMENT'])
                 > 1) OR
                ((SIZEOF (TYPEOF (oa)) > 2) AND
                 (SIZEOF (TYPEOF (oa) *
                        ['AUTOMOTIVE_DESIGN.APPLIED_DATE_AND_ORGANIZATION_ASSIGNMENT',
                         'AUTOMOTIVE_DESIGN.APPLIED_ORGANIZATION_ASSIGNMENT',
                         'AUTOMOTIVE_DESIGN.PROJECT_ASSIGNMENT'])
                  = 1))
```

```
        )) = 0;

END_RULE;
(*
```

Argument definitions:

organization_assignment: the set of all instances of **organization_assignment**.

Formal propositions:

WR1: Each instance of the subtypes of **organization_assignment** shall be one of **applied_date_and_organization_assignment**, **applied_organization_assignment**, or **project_assignment.**

5.2.4.66 person_requires_person_and_organization

The **person_requires_person_and_organization** rule ensures that each instance of **person** shall be referenced by at least one instance of **person_and_organization** as **the_person.** This rule enforces the requirement for each person to have at least one organization specified the person belongs to.

EXPRESS specification:

```
*)
RULE person_requires_person_and_organization FOR
    (person);
WHERE
  WR1:  SIZEOF( QUERY( p <* person |
         SIZEOF( USEDIN(p, 'AUTOMOTIVE_DESIGN.' +
           'PERSON_AND_ORGANIZATION.THE_PERSON')
         ) < 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

person: the set of all instances of **person**.

Formal propositions:

WR1: Each instance of **person** shall be referenced by at least one instance of **person_and_organization** as **the_person.**

5.2.4.67 presentation_layer_assignment_constraint_2d_or_3d

The **presentation_layer_assignment_constraint_2d_or_3d** rule ensures that every element of a layer that is a **point**, **curve**, or **surface** is an element in a **shape_representation** and that every element of a layer that is an **annotation_occurrence** is one of the items in an **draughting_subfigure_representation**, **draughting_model**, **presentation_area**, or **presentation_view**.

EXPRESS specification:

```
*)
RULE presentation_layer_assignment_constraint_2d_or_3d FOR
    (presentation_layer_assignment);
WHERE
  WR1:  SIZEOF (QUERY (pla <* presentation_layer_assignment |
              NOT (SIZEOF (QUERY (item <* pla.assigned_items |
              NOT (SIZEOF (TYPEOF (item) *
                          ['AUTOMOTIVE_DESIGN.POINT',
                           'AUTOMOTIVE_DESIGN.CURVE',
                           'AUTOMOTIVE_DESIGN.SURFACE',
                           'AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE'])=1)))=0
              )))=0;
  WR2:  SIZEOF (QUERY (pla <* presentation_layer_assignment |
              NOT (SIZEOF (QUERY (pnt <* QUERY (item <* pla.assigned_items |
                   ('AUTOMOTIVE_DESIGN.POINT' IN TYPEOF (item))) |
                 NOT (SIZEOF (QUERY (rep <* USEDIN (pnt,
                            'AUTOMOTIVE_DESIGN.REPRESENTATION.ITEMS') |
                       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION'
                             IN TYPEOF (rep)))) = 0)
                   ))=0
              )))=0;
  WR3:  SIZEOF (QUERY (pla <* presentation_layer_assignment |
              NOT (SIZEOF (QUERY (crv <* QUERY (item <* pla.assigned_items |
                   ('AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF (item))) |
                 NOT (SIZEOF (QUERY (rep <* USEDIN (crv,
                            'AUTOMOTIVE_DESIGN.REPRESENTATION.ITEMS') |
                       ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION'
                             IN TYPEOF (rep)))) = 0)
                   ))=0
              )))=0;
  WR4:  SIZEOF (QUERY (pla <* presentation_layer_assignment |
              NOT (SIZEOF (QUERY (srf <* QUERY (item <* pla.assigned_items|
                   ('AUTOMOTIVE_DESIGN.SURFACE' IN TYPEOF (item))) |
                  NOT  (SIZEOF (QUERY (rep <* USEDIN (srf,
                             'AUTOMOTIVE_DESIGN.REPRESENTATION.ITEMS') |
                        ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION'
                              IN TYPEOF (rep)))) = 0)
                   ))=0
              )))=0;
  WR5:  SIZEOF (QUERY (pla <* presentation_layer_assignment |
              NOT (SIZEOF (QUERY (ao <* QUERY (item <* pla.assigned_items|
```

```
                    ('AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE'
                      IN TYPEOF (item))) |
                NOT (SIZEOF (QUERY (ur <* using_representations(ao) |
                  NOT (SIZEOF (TYPEOF(ur) *
                    ['AUTOMOTIVE_DESIGN.' +
                     'DRAUGHTING_SUBFIGURE_REPRESENTATION',
                     'AUTOMOTIVE_DESIGN.DRAUGHTING_MODEL',
                     'AUTOMOTIVE_DESIGN.PRESENTATION_AREA',
                     'AUTOMOTIVE_DESIGN.PRESENTATION_VIEW']) = 1)
                )) = 0)
                ))=0
              )))=0;

END_RULE;
(*
```

Argument definitions:

presentation_layer_assignment: the set of all instances of **presentation_layer_assignment**.

Formal propositions:

WR1: The **assigned_items** of every **presentation_layer_assignment** shall be a **point**, **curve**, **surface**, or **annotation_occurrence.**

WR2: Every **point** that is one of the **assigned_items** of a **presentation_layer_assignment** shall also be a member of the set of **items** of a **shape_representation.**

WR3: Every **curve** that is one of the **assigned_items** of a **presentation_layer_assignment** shall also be a member of the set of **items** of a **shape_representation.**

WR4: Every **surface** that is one of the **assigned_items** of a **presentation_layer_assignment** shall also be a member of the set of **items** of a **shape_representation.**

WR5: Every **annotation_occurrence** that is one of the **assigned_items** of a **presentation_layer_assignment** shall also be a member of the set of **items** of an **draughting_subfigure_representation**, **draughting_model**, **presentation_area**, or **presentation_view.**

5.2.4.68 product_definition_substitute_requires_effectivity_assignment

The **product_definition_substitute_requires_effectivity_assignment** rule specifies that each instance of **product_definition_substitute** shall be referenced by at least one instance of **applied_effective_item_assignment** that specifies the start of the **assigned_effectivity.** This rule enforces the requirement for every instance of replaced usage relationship to have at least one effectivity applied to it specifiying the start of the validity of the replacement.

EXPRESS specification:

```
*)
RULE product_definition_substitute_requires_effectivity_assignment FOR
    (applied_effective_item_assignment,
     product_definition_substitute);
WHERE
  WR1:  SIZEOF( QUERY( pds <* product_definition_substitute |
                SIZEOF( QUERY( aeia <* USEDIN( pds, 'AUTOMOTIVE_DESIGN.' +
                                        'APPLIED_EFFECTIVE_ITEM_ASSIGNMENT.ITEMS') |
                        ('AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY' IN
                         TYPEOF(aeia.assigned_effectivity)) OR
                        (('AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY' IN
                          TYPEOF(aeia.assigned_effectivity)) AND
                         (SIZEOF( QUERY( ea <* USEDIN( aeia.assigned_effectivity,
                                        'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT.ITEMS') |
                                  ea.role.name = 'effectivity start'
                          )) = 1 ))
                )) = 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

applied_effective_item_assignment: the set of all instances of **applied_effective_item_assignment**.

product_definition_substitute: the set of all instances of **product_definition_substitute**.

Formal propositions:

WR1: For each instance of **product_definition_substitute** there shall be at least one instance of **applied_effective_item_assignment** that contains the instance in its set of **items**, and that has an **assigned_effectivity** that is a **dated_effectivity** or that has an **assigned_effectivity** that is an **event_based_effectivity** that is contained in the set of **items** of an **event_assignment** that references an **action_role** with a **name** of 'effectivity start'.

5.2.4.69 product_definition_subtype_exclusiveness

The **product_definition_subtype_exclusiveness** rule specifies that an instance of the subtypes of **product_definition** shall be one of **mated_item_relationship**, **physically_modelled_product_definition**, **product_definition_occurrence**, **product_definition_resource**, or **solution_based_product_definition**.

EXPRESS specification:

```
*)
RULE product_definition_subtype_exclusiveness FOR
    (product_definition);
WHERE
  WR1:  SIZEOF (QUERY (pd <* product_definition |
                (SIZEOF (TYPEOF (pd) *
                         ['AUTOMOTIVE_DESIGN.MATED_ITEM_RELATIONSHIP',
                          'AUTOMOTIVE_DESIGN.PHYSICALLY_MODELLED_PRODUCT_DEFINITION',
                          'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_OCCURRENCE',
                          'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RESOURCE',
                          'AUTOMOTIVE_DESIGN.SOLUTION_BASED_PRODUCT_DEFINITION'])
                 > 1) OR
                ((SIZEOF (TYPEOF (pd)) > 2) AND
                 (SIZEOF (TYPEOF (pd) *
                         ['AUTOMOTIVE_DESIGN.MATED_ITEM_RELATIONSHIP',
                          'AUTOMOTIVE_DESIGN.PHYSICALLY_MODELLED_PRODUCT_DEFINITION',
                          'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_OCCURRENCE',
                          'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RESOURCE',
                          'AUTOMOTIVE_DESIGN.SOLUTION_BASED_PRODUCT_DEFINITION'])
                  = 1))
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_definition: the set of all instances of **product_definition**.

Formal propositions:

WR1: Each instance of the subtypes of **product_definition** shall be one of **mated_item_relationship, physically_modelled_product_definition, product_definition_occurrence, product_definition_resource**, or **solution_based_product_definition.**

5.2.4.70 product_requires_version

The **product_requires_version** rule ensures that each instance of **product** is referenced by at least one instance of **product_definition_formation.** This rule enforces the requirement for every item to have at least one item version associated to it.

EXPRESS specification:

```
*)
RULE product_requires_version FOR
    (product);
WHERE
```

```
  WR1:  SIZEOF( QUERY( p <* product |
         SIZEOF( USEDIN( p, 'AUTOMOTIVE_DESIGN.' +
            'PRODUCT_DEFINITION_FORMATION.OF_PRODUCT')
         ) < 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product: the set of all instances of **product**.

Formal propositions:

WR1: For each instance of **product** there shall be one or more instances of **product_definition_formation** that references it as the **of_product.**

5.2.4.71 representation_subtype_exclusiveness

The **representation_subtype_exclusiveness** rule specifies that an instance of the subtypes of a representation shall be only one of **connected_face_shape_representation**, **constructive_geometry_representation**, **default_tolerance_table**, **default_tolerance_table_cell**, **definitional_representation**, **direction_shape_representation**, **draughting_model**, **externally_defined_representation**, **face_shape_representation**, **kinematic_frame_background_representation** , **kinematic_ground_representation**, **kinematic_link_representation**, **location_shape_representation**, **mechanical_design_geometric_presentation_representation**, **path_shape_representation**, **presentation_representation**, **property_value_representation**, **shape_representation**, **symbol_representation**, or **text_string_representation.**

EXPRESS specification:

```
*)
RULE representation_subtype_exclusiveness FOR
    (representation);
WHERE
  WR1:  SIZEOF (QUERY (pd <* representation |
         (SIZEOF (TYPEOF (pd) *
         ['AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.CONSTRUCTIVE_GEOMETRY_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.DEFAULT_TOLERANCE_TABLE',
          'AUTOMOTIVE_DESIGN.DEFAULT_TOLERANCE_TABLE_CELL',
          'AUTOMOTIVE_DESIGN.DEFINITIONAL_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.DRAUGHTING_MODEL',
          'AUTOMOTIVE_DESIGN.EXTERNALLY_DEFINED_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION',
```

```
          'AUTOMOTIVE_DESIGN.KINEMATIC_FRAME_BACKGROUND_REPRESENTATION ',
          'AUTOMOTIVE_DESIGN.KINEMATIC_GROUND_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.KINEMATIC_LINK_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.LOCATION_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.PATH_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.PRESENTATION_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.PROPERTY_VALUE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.SYMBOL_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.TEXT_STRING_REPRESENTATION'])
                > 1) OR
        ((SIZEOF (TYPEOF (pd)) > 2) AND
        (SIZEOF (TYPEOF (pd) *
        ['AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.CONSTRUCTIVE_GEOMETRY_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.DEFAULT_TOLERANCE_TABLE',
         'AUTOMOTIVE_DESIGN.DEFAULT_TOLERANCE_TABLE_CELL',
         'AUTOMOTIVE_DESIGN.DEFINITIONAL_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.DRAUGHTING_MODEL',
         'AUTOMOTIVE_DESIGN.EXTERNALLY_DEFINED_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.KINEMATIC_FRAME_BACKGROUND_REPRESENTATION ',
         'AUTOMOTIVE_DESIGN.KINEMATIC_GROUND_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.KINEMATIC_LINK_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.LOCATION_SHAPE_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.PATH_SHAPE_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.PRESENTATION_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.PROPERTY_VALUE_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.SYMBOL_REPRESENTATION',
         'AUTOMOTIVE_DESIGN.TEXT_STRING_REPRESENTATION'])
                = 1))
      )) = 0;


END_RULE;
(*
```

Argument definitions:

representation: the set of all instances of **representation**.

Formal propositions:

WR1: Each instance of the subtypes of representation shall be one of **connected_face_shape_representation**, **constructive_geometry_representation**, **default_tolerance_table**, **default_tolerance_table_cell**, **definitional_representation**, **direction_shape_representation**,

draughting_model, **externally_defined_representation**, **face_shape_representation**, **kinematic_frame_background_representation** , **kinematic_ground_representation**, **kinematic_link_representation**, **location_shape_representation**, **mechanical_design_geometric_presentation_representation**, **path_shape_representation**, **presentation_representation**, **property_value_representation**, **shape_representation**, **symbol_representation**, or **text_string_representation**.

5.2.4.72 restrict_action_directive

The **restrict_action_directive** rule specifies that each instance of **action_directive** shall be referenced by at least one instance of **directed_action** as the directive. The only values of the **analysis** of the **action_directive** permitted shall be 'internal' or 'external'. This rule enforces the requirement for every work order to control one or more activities that are performed either internally or externally.

EXPRESS specification:

```
*)
RULE restrict_action_directive FOR
    (action_directive);
WHERE
  WR1:  SIZEOF( QUERY( ad <* action_directive |
                SIZEOF( USEDIN( ad, 'AUTOMOTIVE_DESIGN.DIRECTED_ACTION.DIRECTIVE')
                ) < 1
        )) = 0;
  WR2:  SIZEOF( QUERY( ad <* action_directive |
                 (ad.analysis <> 'internal') AND
                 (ad.analysis <> 'external')
        )) = 0;

END_RULE;
(*
```

Argument definitions:

action_directive: the set of all instances of **action_directive**.

Formal propositions:

WR1: For each instance of **action_directive** there shall be at least one instance of **directed_action** that references the instance of **action_directive** as the directive.

WR2: For each instance of **action_directive** the value of the **analysis** attribute shall be either 'internal' or 'external'.

5.2.4.73 restrict_action_resource_requirement_for_process_operation

The **restrict_action_resource_requirement_for_process_operation** rule specifies that each instance of **action** with a **description** of 'process operation' shall only be referenced by at most one instance of **action_resource_requirement** with a **name** of 'reference tool'. Each instance of **action_resource_requirement** with a **name** of 'reference tool' shall be of type **requirement_for_action_resource** and shall have exactly one item in its set of **resources** that references an instance of **action_resource** which is a **product_definition_resource.** The only **name**s permitted for **action_resource_requirement** shall be 'reference tool' or 'non reference tool'. This rule enforces the requirement for every process operation to have at most one resource as the reference tool associated to it where the reference tool has to be associated to an item version.

EXPRESS specification:

```
*)
RULE restrict_action_resource_requirement_for_process_operation FOR
    (action,
     action_resource_requirement);
WHERE
  WR1:  SIZEOF( QUERY( a <* action |
                (a.description = 'process operation') AND
                (SIZEOF( QUERY( arr <* USEDIN( a, 'AUTOMOTIVE_DESIGN.' +
                                'ACTION_RESOURCE_REQUIREMENT.OPERATIONS') |
                         arr.name = 'reference tool'
                )) > 1 )
        )) = 0;
  WR2:  SIZEOF( QUERY( arr <* action_resource_requirement |
                (arr.name = 'reference tool') AND
                (NOT ('AUTOMOTIVE_DESIGN.REQUIREMENT_FOR_ACTION_RESOURCE' IN
                      TYPEOF(arr))) AND
                (SIZEOF( QUERY( ar <* arr\requirement_for_action_resource.resources |
                         'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RESOURCE' IN
                         TYPEOF(ar)
                 )) <> 1 )
        )) = 0;
  WR3:  SIZEOF( QUERY( arr <* action_resource_requirement |
                (arr.name <> 'reference tool') AND
                (arr.name <> 'non reference tool')
        )) = 0;

END_RULE;
(*
```

Argument definitions:

action: the set of all instances of **action**.

action_resource_requirement: the set of all instances of **action_resource_requirement**.

Formal propositions:

WR1: For each instance of **action** with a **description** of 'process operation' there shall be at most one instance of **action_resource_requirement** with a **name** of 'reference tool' that references the action in its set of operations.

WR2: Each instance of **action_resource_requirement** with a **name** of 'reference tool' shall be a **requirement_for_action_resource** and shall have exactly one item in its set of **resources** that references an instance of **action_resource** which is a **product_definition_resource.**

WR3: For each instance of **action_resource_requirement** the value of the **name** attribute shall be either 'reference tool' or 'non reference tool'.

5.2.4.74 restrict_approval_date_time_for_approval

The **restrict_approval_date_time_for_approval** rule specifies that each instance of **approval** shall be referenced by at most one instance of **approval_date_time** that specifies the actual date of the **approval** and by at most one instance of **approval_date_time** that specifies the nominal date of the **approval**. This rule enforces the requirement for every approval to have at most one actual and at most one nominal date or a nominal date and a time associated to it to indicate when the approval has taken place or will take place.

EXPRESS specification:

```
*)
RULE restrict_approval_date_time_for_approval FOR
    (approval);
WHERE
  WR1:  SIZEOF( QUERY( a <* approval |
         SIZEOF( QUERY( adt <* USEDIN( a, 'AUTOMOTIVE_DESIGN.' +
               'APPROVAL_DATE_TIME.DATED_APPROVAL') |
         adt.role.name = 'actual'
         )) > 1
        )) = 0;
  WR2:  SIZEOF( QUERY( a <* approval |
         SIZEOF( QUERY( adt <* USEDIN( a, 'AUTOMOTIVE_DESIGN.' +
               'APPROVAL_DATE_TIME.DATED_APPROVAL') |
         adt.role.name = 'nominal'
         )) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

approval: the set of all instances of **approval**.

Formal propositions:

WR1: There shall be at most one instance of **approval_date_time** associated to an **approval** where the **name** of **approval_date_time_role** is 'actual'.

WR2: There shall be at most one instance of **approval_date_time** associated to an **approval** where the **name** of **approval_date_time_role** is 'nominal'.

5.2.4.75 restrict_approval_date_time_role

The **restrict_approval_date_time_role** rule specifies that the only values of the **name** of an **approval_date_time_role** permitted shall be 'actual' or 'nominal'.

EXPRESS specification:

```
*)
RULE restrict_approval_date_time_role FOR
    (approval_date_time_role);
WHERE
  WR1:  SIZEOF( QUERY( adtr <* approval_date_time_role |
                (adtr.name <> 'actual') AND
                (adtr.name <> 'nominal')
        )) = 0;

END_RULE;
(*
```

Argument definitions:

approval_date_time_role: the set of all instances of **approval_date_time_role**.

Formal propositions:

WR1: For each instance of **approval_date_time_role** the value of the **name** attribute shall be either 'actual' or 'nominal'.

5.2.4.76 restrict_classification_system_for_approval_status

The **restrict_classification_system_for_approval_status** rule specifies that each instance of **approval_status** shall be referenced by at most one instance of **group_member_assignment.** This rule enforces the requirement for every **approval_status** to have at most one **classification_system** associated to it.

EXPRESS specification:

```
*)
RULE restrict_classification_system_for_approval_status FOR
    (approval_status,
     group_member_assignment);
WHERE
  WR1:  SIZEOF( QUERY( a_s <* approval_status |
                SIZEOF( USEDIN( a_s, 'AUTOMOTIVE_DESIGN.' +
                                     'GROUP_MEMBER_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

approval_status: the set of all instances of **approval_status**.

group_member_assignment: the set of all instances of **group_member_assignment**.

Formal propositions:

WR1: There shall be zero or one instance of **group_member_assignment** associated to an **approval_status.**

5.2.4.77 restrict_classification_system_for_document_type

The **restrict_classification_system_for_document_type** rule specifies that each instance of **document_type** shall be referenced by at most one instance of **group_member_assignment.** This rule enforces the requirement for every document type to have at most one classification system associated to it.

EXPRESS specification:

```
*)
RULE restrict_classification_system_for_document_type FOR
    (document_type,
     group_member_assignment);
WHERE
  WR1:  SIZEOF( QUERY( dt <* document_type |
                SIZEOF( USEDIN( dt, 'AUTOMOTIVE_DESIGN.' +
                                     'GROUP_MEMBER_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

document_type: the set of all instances of **document_type**.

group_member_assignment: the set of all instances of **group_member_assignment**.

Formal propositions:

WR1: There shall be zero or one instance of **group_member_assignment** associated to a **document_type.**

5.2.4.78 restrict_classification_system_for_security_classification_level

The **restrict_classification_system_for_security_classification_level** rule specifies that each instance of **security_classification_level** shall be referenced by at most one instance of **group_member_assignment.** This rule enforces the requirement for every security level to have at most one classification system associated to it.

EXPRESS specification:

```
*)
RULE restrict_classification_system_for_security_classification_level FOR
    (group_member_assignment,
     security_classification_level);
WHERE
  WR1:  SIZEOF( QUERY( scl <* security_classification_level |
                SIZEOF( USEDIN( scl, 'AUTOMOTIVE_DESIGN.' +
                                     'GROUP_MEMBER_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

group_member_assignment: the set of all instances of **group_member_assignment**.

security_classification_level: the set of all instances of **security_classification_level**.

Formal propositions:

WR1: There shall be zero or one instance of **group_member_assignment** associated to a **security_classification_level.**

5.2.4.79 restrict_multi_language_for_action

The **restrict_multi_language_for_action** rule specifies that each instance of **action** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'description' or 'name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_action FOR
    (action);
WHERE
  WR1:  SIZEOF( QUERY( ent <* action |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'description') AND
                        (amlaa.attribute_name <> 'name')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* action |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

action: the set of all instances of action.

Formal propositions:

WR1: Each instance of action shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'description' or 'name'.

WR2: For each instance of action, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.80 restrict_multi_language_for_action_directive

The **restrict_multi_language_for_action_directive** rule specifies that each instance of **action_directive** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'comment'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_action_directive FOR
    (action_directive);
WHERE
  WR1:  SIZEOF( QUERY( ent <* action_directive |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'comment'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* action_directive |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

action_directive: the set of all instances of **action_directive.**

Formal propositions:

WR1: Each instance of **action_directive** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'comment'.

WR2: For each instance of **action_directive,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.81 restrict_multi_language_for_action_method

The **restrict_multi_language_for_action_method** rule specifies that each instance of **action_method** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'description', 'name', 'purpose', or 'consequence'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_action_method FOR
    (action_method);
WHERE
  WR1:  SIZEOF( QUERY( ent <* action_method |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'description') AND
                        (amlaa.attribute_name <> 'name') AND
                        (amlaa.attribute_name <> 'purpose') AND
                        (amlaa.attribute_name <> 'consequence')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* action_method |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                 'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                 (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                 (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

action_method: the set of all instances of **action_method.**

Formal propositions:

WR1: Each instance of **action_method** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'description', 'name', 'purpose', or 'consequence'.

WR2: For each instance of **action_method,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.82 restrict_multi_language_for_action_property

The **restrict_multi_language_for_action_property** rule specifies that each instance of **action_property** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_action_property FOR
    (action_property);
WHERE
  WR1:  SIZEOF( QUERY( ent <* action_property |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* action_property |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

action_property: the set of all instances of **action_property.**

Formal propositions:

WR1: Each instance of **action_property** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **action_property,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.83 restrict_multi_language_for_action_relationship

The **restrict_multi_language_for_action_relationship** rule specifies that each instance of **action_relationship** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_action_relationship FOR
    (action_relationship);
WHERE
  WR1:  SIZEOF( QUERY( ent <* action_relationship |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* action_relationship |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

action_relationship: the set of all instances of **action_relationship.**

Formal propositions:

WR1: Each instance of **action_relationship** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **action_relationship,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.84 restrict_multi_language_for_action_resource_-requirement

The **restrict_multi_language_for_action_resource_requirement** rule specifies that each instance of **action_resource_requirement** shall only be referenced by instances of **applied_-multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_action_resource_requirement FOR
    (action_resource_requirement);
WHERE
  WR1:  SIZEOF( QUERY( ent <* action_resource_requirement |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* action_resource_requirement |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

action_resource_requirement: the set of all instances of **action_resource_requirement.**

Formal propositions:

WR1: Each instance of **action_resource_requirement** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_-name** of 'description'.

WR2: For each instance of **action_resource_requirement,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_-name**.

5.2.4.85 restrict_multi_language_for_alternate_product_relationship

The **restrict_multi_language_for_alternate_product_relationship** rule specifies that each instance of **alternate_product_relationship** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'basis'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_alternate_product_relationship FOR
    (alternate_product_relationship);
WHERE
  WR1:  SIZEOF( QUERY( ent <* alternate_product_relationship |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'basis'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* alternate_product_relationship |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

alternate_product_relationship: the set of all instances of **alternate_product_relationship.**

Formal propositions:

WR1: Each instance of **alternate_product_relationship** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'basis'.

WR2: For each instance of **alternate_product_relationship,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.86 restrict_multi_language_for_application_context

The **restrict_multi_language_for_application_context** rule specifies that each instance of **application_context** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_application_context FOR
    (application_context);
WHERE
  WR1:  SIZEOF( QUERY( ent <* application_context |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* application_context |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

application_context: the set of all instances of **application_context.**

Formal propositions:

WR1: Each instance of **application_context** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **application_context,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.87 restrict_multi_language_for_approval

The **restrict_multi_language_for_approval** rule specifies that each instance of **approval** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'level'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_approval FOR
    (approval);
WHERE
  WR1:  SIZEOF( QUERY( ent <* approval |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'level'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* approval |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

approval: the set of all instances of approval.

Formal propositions:

WR1: Each instance of approval shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'level'.

WR2: For each instance of approval, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.88 restrict_multi_language_for_assembly_component_usage

The **restrict_multi_language_for_assembly_component_usage** rule specifies that each instance of **assembly_component_usage** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'product_definition_relationship.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_assembly_component_usage FOR
    (assembly_component_usage);
WHERE
  WR1:  SIZEOF( QUERY( ent <* assembly_component_usage |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'product_definition_relationship.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* assembly_component_usage |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

assembly_component_usage: the set of all instances of **assembly_component_usage.**

Formal propositions:

WR1: Each instance of **assembly_component_usage** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'product_definition_relationship.description'.

WR2: For each instance of **assembly_component_usage**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.89 restrict_multi_language_for_assembly_component_usage_substitute

The **restrict_multi_language_for_assembly_component_usage_substitute** rule specifies that each instance of **assembly_component_usage_substitute** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_assembly_component_usage_substitute FOR
    (assembly_component_usage_substitute);
WHERE
  WR1:  SIZEOF( QUERY( ent <* assembly_component_usage_substitute |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* assembly_component_usage_substitute |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

assembly_component_usage_substitute: the set of all instances of **assembly_component_usage_substitute.**

Formal propositions:

WR1: Each instance of **assembly_component_usage_substitute** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **assembly_component_usage_substitute,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.90 restrict_multi_language_for_camera_image_2d_with_-scale

The **restrict_multi_language_for_camera_image_2d_with_scale** rule specifies that each instance of **camera_image_2d_with_scale** shall only be referenced by instances of **applied_-multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation_item.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_camera_image_2d_with_scale FOR
    (camera_image_2d_with_scale);
WHERE
  WR1:  SIZEOF( QUERY( ent <* camera_image_2d_with_scale |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'representation_item.name'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* camera_image_2d_with_scale |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

camera_image_2d_with_scale: the set of all instances of **camera_image_2d_with_scale.**

Formal propositions:

WR1: Each instance of **camera_image_2d_with_scale** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation_item.name'.

WR2: For each instance of **camera_image_2d_with_scale,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_-name**.

5.2.4.91 restrict_multi_language_for_camera_image_3d_with_-scale

The **restrict_multi_language_for_camera_image_3d_with_scale** rule specifies that each instance of **camera_image_3d_with_scale** shall only be referenced by instances of **applied_-multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation_item.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_camera_image_3d_with_scale FOR
    (camera_image_3d_with_scale);
WHERE
  WR1:  SIZEOF( QUERY( ent <* camera_image_3d_with_scale |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'representation_item.name'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* camera_image_3d_with_scale |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

camera_image_3d_with_scale: the set of all instances of **camera_image_3d_with_scale.**

Formal propositions:

WR1: Each instance of **camera_image_3d_with_scale** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation_item.name'.

WR2: For each instance of **camera_image_3d_with_scale,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_-name**.

5.2.4.92 restrict_multi_language_for_certification

The **restrict_multi_language_for_certification** rule specifies that each instance of **certification** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'name' or 'purpose'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_certification FOR
    (certification);
WHERE
  WR1:  SIZEOF( QUERY( ent <* certification |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'name') AND
                        (amlaa.attribute_name <> 'purpose')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* certification |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

certification: the set of all instances of certification.

Formal propositions:

WR1: Each instance of certification shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'name' or 'purpose'.

WR2: For each instance of certification, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.93 restrict_multi_language_for_characterized_product_category

The **restrict_multi_language_for_characterized_product_category** rule specifies that each instance of **characterized_product_category** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'product_category.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_characterized_product_category FOR
    (characterized_product_category);
WHERE
  WR1:  SIZEOF( QUERY( ent <* characterized_product_category |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'product_category.name'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* characterized_product_category |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

characterized_product_category: the set of all instances of **characterized_product_category.**

Formal propositions:

WR1: Each instance of **characterized_product_category** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'product_category.name'.

WR2: For each instance of **characterized_product_category,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.94 restrict_multi_language_for_classification_system

The **restrict_multi_language_for_classification_system** rule specifies that each instance of **classification_system** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'group.description' or 'group.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_classification_system FOR
    (classification_system);
WHERE
  WR1:  SIZEOF( QUERY( ent <* classification_system |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'group.description') AND
                        (amlaa.attribute_name <> 'group.name')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* classification_system |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

classification_system: the set of all instances of **classification_system.**

Formal propositions:

WR1: Each instance of **classification_system** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'group.description' or 'group.name'.

WR2: For each instance of **classification_system**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.95 restrict_multi_language_for_configurable_item

The **restrict_multi_language_for_configurable_item** rule specifies that each instance of **configurable_item** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'configuration_item.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_configurable_item FOR
    (configurable_item);
WHERE
  WR1:  SIZEOF( QUERY( ent <* configurable_item |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'configuration_item.name'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* configurable_item |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

configurable_item: the set of all instances of **configurable_item.**

Formal propositions:

WR1: Each instance of **configurable_item** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'configuration_item.name'.

WR2: For each instance of **configurable_item,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.96 restrict_multi_language_for_configuration_design

The **restrict_multi_language_for_configuration_design** rule specifies that each instance of **configuration_design** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_configuration_design FOR
    (configuration_design);
WHERE
  WR1:  SIZEOF( QUERY( ent <* configuration_design |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* configuration_design |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

configuration_design: the set of all instances of **configuration_design.**

Formal propositions:

WR1: Each instance of **configuration_design** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **configuration_design,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.97 restrict_multi_language_for_configuration_item_relationship

The **restrict_multi_language_for_configuration_item_relationship** rule specifies that each instance of **configuration_item_relationship** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_configuration_item_relationship FOR
    (configuration_item_relationship);
WHERE
  WR1:  SIZEOF( QUERY( ent <* configuration_item_relationship |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* configuration_item_relationship |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

configuration_item_relationship: the set of all instances of **configuration_item_relationship.**

Formal propositions:

WR1: Each instance of **configuration_item_relationship** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **configuration_item_relationship,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.98 restrict_multi_language_for_constructive_geometry_representation_relationship

The **restrict_multi_language_for_constructive_geometry_representation_relationship** rule specifies that each instance of **constructive_geometry_representation_relationship** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation_relationship.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_constructive_geometry_representation_relationship FOR
    (constructive_geometry_representation_relationship);
WHERE
  WR1:  SIZEOF( QUERY( ent <* constructive_geometry_representation_relationship |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'representation_relationship.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* constructive_geometry_representation_relationship |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

constructive_geometry_representation_relationship: the set of all instances of **constructive_geometry_representation_relationship.**

Formal propositions:

WR1: Each instance of **constructive_geometry_representation_relationship** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation_relationship.description'.

WR2: For each instance of **constructive_geometry_representation_relationship,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.99 restrict_multi_language_for_data_environment

The **restrict_multi_language_for_data_environment** rule specifies that each instance of **data_environment** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_data_environment FOR
    (data_environment);
WHERE
  WR1:  SIZEOF( QUERY( ent <* data_environment |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* data_environment |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

data_environment: the set of all instances of **data_environment.**

Formal propositions:

WR1: Each instance of **data_environment** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **data_environment,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.100 restrict_multi_language_for_default_tolerance_table_cell

The **restrict_multi_language_for_default_tolerance_table_cell** rule specifies that each instance of **default_tolerance_table_cell** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_default_tolerance_table_cell FOR
    (default_tolerance_table_cell);
WHERE
  WR1:  SIZEOF( QUERY( ent <* default_tolerance_table_cell |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'representation.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* default_tolerance_table_cell |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

default_tolerance_table_cell: the set of all instances of **default_tolerance_table_cell.**

Formal propositions:

WR1: Each instance of **default_tolerance_table_cell** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation.description'.

WR2: For each instance of **default_tolerance_table_cell,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.101 restrict_multi_language_for_descriptive_representation_item

The **restrict_multi_language_for_descriptive_representation_item** rule specifies that each instance of **descriptive_representation_item** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_descriptive_representation_item FOR
    (descriptive_representation_item);
WHERE
  WR1:  SIZEOF( QUERY( ent <* descriptive_representation_item |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* descriptive_representation_item |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

descriptive_representation_item: the set of all instances of **descriptive_representation_item.**

Formal propositions:

WR1: Each instance of **descriptive_representation_item** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **descriptive_representation_item**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.102 restrict_multi_language_for_dimensional_location

The **restrict_multi_language_for_dimensional_location** rule specifies that each instance of **dimensional_location** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'shape_aspect_relationship.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_dimensional_location FOR
    (dimensional_location);
WHERE
  WR1:  SIZEOF( QUERY( ent <* dimensional_location |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'shape_aspect_relationship.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* dimensional_location |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                 'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                 (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                 (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

dimensional_location: the set of all instances of **dimensional_location.**

Formal propositions:

WR1: Each instance of **dimensional_location** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'shape_aspect_relationship.description'.

WR2: For each instance of **dimensional_location,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.103 restrict_multi_language_for_dimensional_size

The **restrict_multi_language_for_dimensional_size** rule specifies that each instance of **dimensional_size** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'applies_to.shape_aspect.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_dimensional_size FOR
    (dimensional_size);
WHERE
  WR1:  SIZEOF( QUERY( ent <* dimensional_size |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'applies_to.shape_aspect.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* dimensional_size |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

dimensional_size: the set of all instances of **dimensional_size.**

Formal propositions:

WR1: Each instance of **dimensional_size** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'applies_to.shape_aspect.description'.

WR2: For each instance of **dimensional_size,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.104 restrict_multi_language_for_document

The **restrict_multi_language_for_document** rule specifies that each instance of **document** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'description' or 'name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_document FOR
    (document);
WHERE
  WR1:  SIZEOF( QUERY( ent <* document |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'description') AND
                        (amlaa.attribute_name <> 'name')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* document |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

document: the set of all instances of document.

Formal propositions:

WR1: Each instance of document shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'description' or 'name'.

WR2: For each instance of document, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.105 restrict_multi_language_for_document_relationship

The **restrict_multi_language_for_document_relationship** rule specifies that each instance of **document_relationship** shall only be referenced by instances of **applied_multi_language_-attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_document_relationship FOR
    (document_relationship);
WHERE
  WR1:  SIZEOF( QUERY( ent <* document_relationship |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* document_relationship |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

document_relationship: the set of all instances of **document_relationship.**

Formal propositions:

WR1: Each instance of **document_relationship** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **document_relationship**, there shall not be more than one **multi_-language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.106 restrict_multi_language_for_draughting_model

The **restrict_multi_language_for_draughting_model** rule specifies that each instance of **draughting_model** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_draughting_model FOR
    (draughting_model);
WHERE
  WR1:  SIZEOF( QUERY( ent <* draughting_model |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'representation.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* draughting_model |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                 'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                 (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                 (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

draughting_model: the set of all instances of **draughting_model.**

Formal propositions:

WR1: Each instance of **draughting_model** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation.description'.

WR2: For each instance of **draughting_model,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.107 restrict_multi_language_for_draughting_title

The **restrict_multi_language_for_draughting_title** rule specifies that each instance of **draughting_title** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'contents'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_draughting_title FOR
    (draughting_title);
WHERE
  WR1:  SIZEOF( QUERY( ent <* draughting_title |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'contents'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* draughting_title |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

draughting_title: the set of all instances of **draughting_title.**

Formal propositions:

WR1: Each instance of **draughting_title** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'contents'.

WR2: For each instance of **draughting_title**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.108 restrict_multi_language_for_drawing_sheet_revision

The **restrict_multi_language_for_drawing_sheet_revision** rule specifies that each instance of **drawing_sheet_revision** shall only be referenced by instances of **applied_multi_language_-**

attribute_assignment that have an attribute **attribute_name** of 'representation.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_drawing_sheet_revision FOR
    (drawing_sheet_revision);
WHERE
  WR1:  SIZEOF( QUERY( ent <* drawing_sheet_revision |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'representation.name'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* drawing_sheet_revision |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

drawing_sheet_revision: the set of all instances of **drawing_sheet_revision.**

Formal propositions:

WR1: Each instance of **drawing_sheet_revision** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation.name'.

WR2: For each instance of **drawing_sheet_revision**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.109 restrict_multi_language_for_effectivity

The **restrict_multi_language_for_effectivity** rule specifies that each instance of **effectivity** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_effectivity FOR
    (effectivity);
WHERE
  WR1:  SIZEOF( QUERY( ent <* effectivity |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* effectivity |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

effectivity: the set of all instances of effectivity.

Formal propositions:

WR1: Each instance of effectivity shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of effectivity, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.110 restrict_multi_language_for_effectivity_relationship

The **restrict_multi_language_for_effectivity_relationship** rule specifies that each instance of **effectivity_relationship** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_effectivity_relationship FOR
    (effectivity_relationship);
```

```
WHERE
  WR1:  SIZEOF( QUERY( ent <* effectivity_relationship |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* effectivity_relationship |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

effectivity_relationship: the set of all instances of **effectivity_relationship.**

Formal propositions:

WR1: Each instance of **effectivity_relationship** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **effectivity_relationship,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.111 restrict_multi_language_for_externally_defined_classification_attribute

The **restrict_multi_language_for_externally_defined_classification_attribute** rule specifies that each instance of **externally_defined_classification_attribute** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_externally_defined_classification_attribute FOR
    (externally_defined_classification_attribute);
WHERE
```

```
  WR1:  SIZEOF( QUERY( ent <* externally_defined_classification_attribute |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'representation.name'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* externally_defined_classification_attribute |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                 'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                 (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                 (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

externally_defined_classification_attribute: the set of all instances of **externally_defined_classification_attribute.**

Formal propositions:

WR1: Each instance of **externally_defined_classification_attribute** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation.name'.

WR2: For each instance of **externally_defined_classification_attribute,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.112 restrict_multi_language_for_externally_defined_classification_system

The **restrict_multi_language_for_externally_defined_classification_system** rule specifies that each instance of **externally_defined_classification_system** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'group.description' or 'group.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_externally_defined_classification_system FOR
    (externally_defined_classification_system);
```

```
WHERE
  WR1:  SIZEOF( QUERY( ent <* externally_defined_classification_system |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'group.description') AND
                        (amlaa.attribute_name <> 'group.name')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* externally_defined_classification_system |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

externally_defined_classification_system: the set of all instances of **externally_defined_classification_system.**

Formal propositions:

WR1: Each instance of **externally_defined_classification_system** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'group.description' or 'group.name'.

WR2: For each instance of **externally_defined_classification_system,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.113 restrict_multi_language_for_externally_defined_product_category

The **restrict_multi_language_for_externally_defined_product_category** rule specifies that each instance of **externally_defined_product_category** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'product_category.name'.

EXPRESS specification:

```
*)
```

```
RULE restrict_multi_language_for_externally_defined_product_category FOR
    (externally_defined_product_category);
WHERE
  WR1:  SIZEOF( QUERY( ent <* externally_defined_product_category |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'product_category.name'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* externally_defined_product_category |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                 'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                 (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                 (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

externally_defined_product_category: the set of all instances of **externally_defined_product_category.**

Formal propositions:

WR1: Each instance of **externally_defined_product_category** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'product_category.name'.

WR2: For each instance of **externally_defined_product_category,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.114 restrict_multi_language_for_externally_defined_representation

The **restrict_multi_language_for_externally_defined_representation** rule specifies that each instance of **externally_defined_representation** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_externally_defined_representation FOR
    (externally_defined_representation);
WHERE
  WR1:  SIZEOF( QUERY( ent <* externally_defined_representation |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* externally_defined_representation |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

externally_defined_representation: the set of all instances of **externally_defined_representation.**

Formal propositions:

WR1: Each instance of **externally_defined_representation** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **externally_defined_representation,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.115 restrict_multi_language_for_geometric_tolerance

The **restrict_multi_language_for_geometric_tolerance** rule specifies that each instance of **geometric_tolerance** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'name' or 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_geometric_tolerance FOR
    (geometric_tolerance);
WHERE
  WR1:  SIZEOF( QUERY( ent <* geometric_tolerance |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'name') AND
                        (amlaa.attribute_name <> 'description')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* geometric_tolerance |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

geometric_tolerance: the set of all instances of **geometric_tolerance.**

Formal propositions:

WR1: Each instance of **geometric_tolerance** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'name' or 'description'.

WR2: For each instance of **geometric_tolerance,** there shall not be more than one **multi_-language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.116 restrict_multi_language_for_group

The **restrict_multi_language_for_group** rule specifies that each instance of **group** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'name' or 'description'.

EXPRESS specification:

```
*)
```

```
RULE restrict_multi_language_for_group FOR
    (group);
WHERE
  WR1:  SIZEOF( QUERY( ent <* group |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'name') AND
                        (amlaa.attribute_name <> 'description')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* group |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

group: the set of all instances of group.

Formal propositions:

WR1: Each instance of group shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'name' or 'description'.

WR2: For each instance of group, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.117 restrict_multi_language_for_identification_role

The **restrict_multi_language_for_identification_role** rule specifies that each instance of **identification_role** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_identification_role FOR
    (identification_role);
WHERE
```

```
  WR1:  SIZEOF( QUERY( ent <* identification_role |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* identification_role |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                 'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                 (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                 (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

identification_role: the set of all instances of **identification_role.**

Formal propositions:

WR1: Each instance of **identification_role** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **identification_role**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.118 restrict_multi_language_for_make_from_usage_option

The **restrict_multi_language_for_make_from_usage_option** rule specifies that each instance of **make_from_usage_option** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'product_definition_relationship.description' or 'product_definition_relationship.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_make_from_usage_option FOR
    (make_from_usage_option);
WHERE
  WR1:  SIZEOF( QUERY( ent <* make_from_usage_option |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                       'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
```

```
        (amlaa.attribute_name <> 'product_definition_relationship.description') AND
         (amlaa.attribute_name <> 'product_definition_relationship.name')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* make_from_usage_option |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                      'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                       'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

make_from_usage_option: the set of all instances of **make_from_usage_option.**

Formal propositions:

WR1: Each instance of **make_from_usage_option** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'product_definition_relationship.description' or 'product_definition_relationship.name'.

WR2: For each instance of **make_from_usage_option**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.119 restrict_multi_language_for_material_property

The **restrict_multi_language_for_material_property** rule specifies that each instance of **material_property** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'property_definition.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_material_property FOR
    (material_property);
WHERE
  WR1:  SIZEOF( QUERY( ent <* material_property |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'property_definition.description'
                )) > 0
```

```
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* material_property |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

material_property: the set of all instances of **material_property.**

Formal propositions:

WR1: Each instance of **material_property** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'property_definition.description'.

WR2: For each instance of **material_property,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.120 restrict_multi_language_for_measure_representation_item

The **restrict_multi_language_for_measure_representation_item** rule specifies that each instance of **measure_representation_item** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation_item.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_measure_representation_item FOR
    (measure_representation_item);
WHERE
  WR1:  SIZEOF( QUERY( ent <* measure_representation_item |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'representation_item.name'
                )) > 0
        )) = 0;
```

```
  WR2:  SIZEOF( QUERY( ent <* measure_representation_item |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

measure_representation_item: the set of all instances of **measure_representation_item.**

Formal propositions:

WR1: Each instance of **measure_representation_item** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation_item.name'.

WR2: For each instance of **measure_representation_item**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.121 restrict_multi_language_for_mechanical_design_geometric_presentation_representation

The **restrict_multi_language_for_mechanical_design_geometric_presentation_representation** rule specifies that each instance of **mechanical_design_geometric_presentation_representation** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation.description'.

EXPRESS specification:

```
*)
RULE
restrict_multi_language_for_mechanical_design_geometric_presentation_representation
 FOR (mechanical_design_geometric_presentation_representation);
WHERE
  WR1:  SIZEOF( QUERY( ent <* mechanical_design_geometric_presentation_representation |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'representation.description'
                )) > 0
```

```
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* mechanical_design_geometric_presentation_representation |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

mechanical_design_geometric_presentation_representation: the set of all instances of **mechanical_design_geometric_presentation_representation.**

Formal propositions:

WR1: Each instance of **mechanical_design_geometric_presentation_representation** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation.description'.

WR2: For each instance of **mechanical_design_geometric_presentation_representation**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.122 restrict_multi_language_for_model_change

The **restrict_multi_language_for_model_change** rule specifies that each instance of **model_change** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'property_definition.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_model_change FOR
    (model_change);
WHERE
  WR1:  SIZEOF( QUERY( ent <* model_change |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'property_definition.description'
                )) > 0
        )) = 0;
```

```
  WR2:  SIZEOF( QUERY( ent <* model_change |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

model_change: the set of all instances of **model_change.**

Formal propositions:

WR1: Each instance of **model_change** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'property_definition.description'.

WR2: For each instance of **model_change,** there shall not be more than one **multi_language_-attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.123 restrict_multi_language_for_organization_relationship

The **restrict_multi_language_for_organization_relationship** rule specifies that each instance of **organization_relationship** shall only be referenced by instances of **applied_multi_language_-attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_organization_relationship FOR
    (organization_relationship);
WHERE
  WR1:  SIZEOF( QUERY( ent <* organization_relationship |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* organization_relationship |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
```

```
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

organization_relationship: the set of all instances of **organization_relationship.**

Formal propositions:

WR1: Each instance of **organization_relationship** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **organization_relationship,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.124 restrict_multi_language_for_organization_role

The **restrict_multi_language_for_organization_role** rule specifies that each instance of **organization_role** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_organization_role FOR
    (organization_role);
WHERE
  WR1:  SIZEOF( QUERY( ent <* organization_role |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'name'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* organization_role |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
```

```
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

organization_role: the set of all instances of **organization_role.**

Formal propositions:

WR1: Each instance of **organization_role** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'name'.

WR2: For each instance of **organization_role,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.125 restrict_multi_language_for_pair_actuator

The **restrict_multi_language_for_pair_actuator** rule specifies that each instance of **pair_actuator** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_pair_actuator FOR
    (pair_actuator);
WHERE
  WR1:  SIZEOF( QUERY( ent <* pair_actuator |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'name'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* pair_actuator |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
```

```
(*
```

Argument definitions:

pair_actuator: the set of all instances of **pair_actuator.**

Formal propositions:

WR1: Each instance of **pair_actuator** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'name'.

WR2: For each instance of **pair_actuator**, there shall not be more than one **multi_language_-attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.126 restrict_multi_language_for_presentation_area

The **restrict_multi_language_for_presentation_area** rule specifies that each instance of **presentation_area** shall only be referenced by instances of **applied_multi_language_attribute_-assignment** that have an attribute **attribute_name** of either 'representation.description' or 'representation.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_presentation_area FOR
    (presentation_area);
WHERE
  WR1:  SIZEOF( QUERY( ent <* presentation_area |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'representation.description') AND
                        (amlaa.attribute_name <> 'representation.name')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* presentation_area |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

presentation_area: the set of all instances of **presentation_area.**

Formal propositions:

WR1: Each instance of **presentation_area** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'representation.description' or 'representation.name'.

WR2: For each instance of **presentation_area,** there shall not be more than one **multi_-language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.127 restrict_multi_language_for_presentation_layer_assignment

The **restrict_multi_language_for_presentation_layer_assignment** rule specifies that each instance of **presentation_layer_assignment** shall only be referenced by instances of **applied_-multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_presentation_layer_assignment FOR
    (presentation_layer_assignment);
WHERE
  WR1:  SIZEOF( QUERY( ent <* presentation_layer_assignment |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* presentation_layer_assignment |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

presentation_layer_assignment: the set of all instances of **presentation_layer_assignment.**

Formal propositions:

WR1: Each instance of **presentation_layer_assignment** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **presentation_layer_assignment,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.128 restrict_multi_language_for_presentation_view

The **restrict_multi_language_for_presentation_view** rule specifies that each instance of **presentation_view** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'representation.description' or 'representation.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_presentation_view FOR
    (presentation_view);
WHERE
  WR1:  SIZEOF( QUERY( ent <* presentation_view |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'representation.description') AND
                        (amlaa.attribute_name <> 'representation.name')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* presentation_view |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

presentation_view: the set of all instances of **presentation_view.**

Formal propositions:

WR1: Each instance of **presentation_view** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'representation.description' or 'representation.name'.

WR2: For each instance of **presentation_view,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.129 restrict_multi_language_for_presentation_view_relationship

The **restrict_multi_language_for_presentation_view_relationship** rule specifies that each instance of **presentation_view_relationship** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation_relationship.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_presentation_view_relationship FOR
    (presentation_view_relationship);
WHERE
  WR1:  SIZEOF( QUERY( ent <* presentation_view_relationship |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'representation_relationship.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* presentation_view_relationship |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

presentation_view_relationship: the set of all instances of **presentation_view_relationship.**

Formal propositions:

WR1: Each instance of **presentation_view_relationship** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation_relationship.description'.

WR2: For each instance of **presentation_view_relationship,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.130 restrict_multi_language_for_process_change

The **restrict_multi_language_for_process_change** rule specifies that each instance of **process_change** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'action_property.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_process_change FOR
    (process_change);
WHERE
  WR1:  SIZEOF( QUERY( ent <* process_change |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'action_property.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* process_change |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

process_change: the set of all instances of **process_change.**

Formal propositions:

WR1: Each instance of **process_change** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'action_property.description'.

WR2: For each instance of **process_change**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.131 restrict_multi_language_for_process_plan

The **restrict_multi_language_for_process_plan** rule specifies that each instance of **process_plan** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'action.description' or 'action.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_process_plan FOR
    (process_plan);
WHERE
  WR1:  SIZEOF( QUERY( ent <* process_plan |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'action.description') AND
                        (amlaa.attribute_name <> 'action.name')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* process_plan |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

process_plan: the set of all instances of **process_plan.**

Formal propositions:

WR1: Each instance of **process_plan** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'action.description' or 'action.name'.

WR2: For each instance of **process_plan,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.132 restrict_multi_language_for_process_product_association

The **restrict_multi_language_for_process_product_association** rule specifies that each instance of **process_product_association** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_process_product_association FOR
    (process_product_association);
WHERE
  WR1:  SIZEOF( QUERY( ent <* process_product_association |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* process_product_association |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

process_product_association: the set of all instances of **process_product_association.**

Formal propositions:

WR1: Each instance of **process_product_association** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **process_product_association,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.133 restrict_multi_language_for_product

The **restrict_multi_language_for_product** rule specifies that each instance of **product** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'description' or 'name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product FOR
    (product);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'description') AND
                        (amlaa.attribute_name <> 'name')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product: the set of all instances of product.

Formal propositions:

WR1: Each instance of product shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'description' or 'name'.

WR2: For each instance of product, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.134 restrict_multi_language_for_product_category

The **restrict_multi_language_for_product_category** rule specifies that each instance of **product_category** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product_category FOR
    (product_category);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product_category |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product_category |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_category: the set of all instances of **product_category.**

Formal propositions:

WR1: Each instance of **product_category** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'des-

cription'.

WR2: For each instance of **product_category**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.135 restrict_multi_language_for_product_class

The **restrict_multi_language_for_product_class** rule specifies that each instance of **product_class** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'product_concept.description' or 'product_concept.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product_class FOR
    (product_class);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product_class |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'product_concept.description') AND
                        (amlaa.attribute_name <> 'product_concept.name')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product_class |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_class: the set of all instances of **product_class.**

Formal propositions:

WR1: Each instance of **product_class** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'product_concept.description' or 'product_concept.name'.

WR2: For each instance of **product_class**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.136 restrict_multi_language_for_product_component

The **restrict_multi_language_for_product_component** rule specifies that each instance of **product_component** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'configuration_item.description' or 'configuration_item.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product_component FOR
    (product_component);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product_component |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'configuration_item.description') AND
                        (amlaa.attribute_name <> 'configuration_item.name')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product_component |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_component: the set of all instances of **product_component.**

Formal propositions:

WR1: Each instance of **product_component** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'configuration_item.description' or 'configuration_item.name'.

WR2: For each instance of **product_component**, there shall not be more than one **multi_-language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.137 restrict_multi_language_for_product_concept_feature

The **restrict_multi_language_for_product_concept_feature** rule specifies that each instance of **product_concept_feature** shall only be referenced by instances of **applied_multi_language_-attribute_assignment** that have an attribute **attribute_name** of either 'description' or 'name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product_concept_feature FOR
    (product_concept_feature);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product_concept_feature |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'description') AND
                        (amlaa.attribute_name <> 'name')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product_concept_feature |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_concept_feature: the set of all instances of **product_concept_feature.**

Formal propositions:

WR1: Each instance of **product_concept_feature** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'description' or 'name'.

WR2: For each instance of **product_concept_feature**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_-**

name.

5.2.4.138 restrict_multi_language_for_product_concept_feature_-association

The **restrict_multi_language_for_product_concept_feature_association** rule specifies that each instance of **product_concept_feature_association** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product_concept_feature_association FOR
    (product_concept_feature_association);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product_concept_feature_association |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product_concept_feature_association |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                 'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                 (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                 (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_concept_feature_association: the set of all instances of **product_concept_feature_association.**

Formal propositions:

WR1: Each instance of **product_concept_feature_association** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **product_concept_feature_association**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.139 restrict_multi_language_for_product_concept_feature_category

The **restrict_multi_language_for_product_concept_feature_category** rule specifies that each instance of **product_concept_feature_category** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'group.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product_concept_feature_category FOR
    (product_concept_feature_category);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product_concept_feature_category |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'group.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product_concept_feature_category |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_concept_feature_category: the set of all instances of **product_concept_feature_category.**

Formal propositions:

WR1: Each instance of **product_concept_feature_category** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attri-**

bute_name of 'group.description'.

WR2: For each instance of **product_concept_feature_category,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.140 restrict_multi_language_for_product_definition

The **restrict_multi_language_for_product_definition** rule specifies that each instance of **product_definition** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'name' or 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product_definition FOR
    (product_definition);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product_definition |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'name') AND
                        (amlaa.attribute_name <> 'description')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product_definition |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_definition: the set of all instances of **product_definition.**

Formal propositions:

WR1: Each instance of **product_definition** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'name' or 'description'.

WR2: For each instance of **product_definition**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.141 restrict_multi_language_for_product_definition_formation

The **restrict_multi_language_for_product_definition_formation** rule specifies that each instance of **product_definition_formation** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product_definition_formation FOR
    (product_definition_formation);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product_definition_formation |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product_definition_formation |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                 'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                 (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                 (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_definition_formation: the set of all instances of **product_definition_formation.**

Formal propositions:

WR1: Each instance of **product_definition_formation** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **product_definition_formation,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.142 restrict_multi_language_for_product_definition_formation_relationship

The **restrict_multi_language_for_product_definition_formation_relationship** rule specifies that each instance of **product_definition_formation_relationship** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product_definition_formation_relationship FOR
    (product_definition_formation_relationship);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product_definition_formation_relationship |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product_definition_formation_relationship |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_definition_formation_relationship: the set of all instances of **product_definition_formation_relationship.**

Formal propositions:

WR1: Each instance of **product_definition_formation_relationship** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute

attribute_name of 'description'.

WR2: For each instance of **product_definition_formation_relationship,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.143 restrict_multi_language_for_product_definition_occurrence

The **restrict_multi_language_for_product_definition_occurrence** rule specifies that each instance of **product_definition_occurrence** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'product_definition.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product_definition_occurrence FOR
    (product_definition_occurrence);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product_definition_occurrence |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'product_definition.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product_definition_occurrence |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_definition_occurrence: the set of all instances of **product_definition_occurrence.**

Formal propositions:

WR1: Each instance of **product_definition_occurrence** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'product_definition.description'.

WR2: For each instance of **product_definition_occurrence,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.144 restrict_multi_language_for_product_definition_relationship

The **restrict_multi_language_for_product_definition_relationship** rule specifies that each instance of **product_definition_relationship** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'description' or 'id'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product_definition_relationship FOR
    (product_definition_relationship);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product_definition_relationship |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'description') AND
                        (amlaa.attribute_name <> 'id')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product_definition_relationship |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_definition_relationship: the set of all instances of **product_definition_relationship.**

Formal propositions:

WR1: Each instance of **product_definition_relationship** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'description' or 'id'.

WR2: For each instance of **product_definition_relationship,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.145 restrict_multi_language_for_product_definition_shape

The **restrict_multi_language_for_product_definition_shape** rule specifies that each instance of **product_definition_shape** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'property_definition.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product_definition_shape FOR
    (product_definition_shape);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product_definition_shape |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'property_definition.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product_definition_shape |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_definition_shape: the set of all instances of **product_definition_shape.**

Formal propositions:

WR1: Each instance of **product_definition_shape** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'property_definition.description'.

WR2: For each instance of **product_definition_shape**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.146 restrict_multi_language_for_product_definition_substitute

The **restrict_multi_language_for_product_definition_substitute** rule specifies that each instance of **product_definition_substitute** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_product_definition_substitute FOR
    (product_definition_substitute);
WHERE
  WR1:  SIZEOF( QUERY( ent <* product_definition_substitute |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* product_definition_substitute |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_definition_substitute: the set of all instances of **product_definition_substitute.**

Formal propositions:

WR1: Each instance of **product_definition_substitute** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **product_definition_substitute,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.147 restrict_multi_language_for_project

The **restrict_multi_language_for_project** rule specifies that each instance of **project** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'organization.description' or 'organization.name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_project FOR
    (project);
WHERE
  WR1:  SIZEOF( QUERY( ent <* project |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'organization.description') AND
                        (amlaa.attribute_name <> 'organization.name')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* project |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

project: the set of all instances of project.

Formal propositions:

WR1: Each instance of project shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'organization.description' or 'organization.name'.

WR2: For each instance of project, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.148 restrict_multi_language_for_property_change

The **restrict_multi_language_for_property_change** rule specifies that each instance of **property_change** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'property_definition.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_property_change FOR
    (property_change);
WHERE
  WR1:  SIZEOF( QUERY( ent <* property_change |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'property_definition.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* property_change |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

property_change: the set of all instances of **property_change.**

Formal propositions:

WR1: Each instance of **property_change** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'property_definition.description'.

WR2: For each instance of **property_change**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.149 restrict_multi_language_for_property_definition

The **restrict_multi_language_for_property_definition** rule specifies that each instance of **property_definition** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_property_definition FOR
    (property_definition);
WHERE
  WR1:  SIZEOF( QUERY( ent <* property_definition |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* property_definition |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                 'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                 (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                 (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

property_definition: the set of all instances of **property_definition.**

Formal propositions:

WR1: Each instance of **property_definition** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'des-

cription'.

WR2: For each instance of **property_definition,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.150 restrict_multi_language_for_property_definition_relationship

The **restrict_multi_language_for_property_definition_relationship** rule specifies that each instance of **property_definition_relationship** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_property_definition_relationship FOR
    (property_definition_relationship);
WHERE
  WR1:  SIZEOF( QUERY( ent <* property_definition_relationship |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* property_definition_relationship |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

property_definition_relationship: the set of all instances of **property_definition_relationship.**

Formal propositions:

WR1: Each instance of **property_definition_relationship** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_-**

name of 'description'.

WR2: For each instance of **property_definition_relationship,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.151 restrict_multi_language_for_property_definition_representation

The **restrict_multi_language_for_property_definition_representation** rule specifies that each instance of **property_definition_representation** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_property_definition_representation FOR
    (property_definition_representation);
WHERE
  WR1:  SIZEOF( QUERY( ent <* property_definition_representation |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'name'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* property_definition_representation |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

property_definition_representation: the set of all instances of **property_definition_representation.**

Formal propositions:

WR1: Each instance of **property_definition_representation** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'name'.

WR2: For each instance of **property_definition_representation,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.152 restrict_multi_language_for_rep_item_group

The **restrict_multi_language_for_rep_item_group** rule specifies that each instance of **rep_item_group** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'group.name' or 'group.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_rep_item_group FOR
    (rep_item_group);
WHERE
  WR1:  SIZEOF( QUERY( ent <* rep_item_group |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'group.name') AND
                        (amlaa.attribute_name <> 'group.description')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* rep_item_group |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

rep_item_group: the set of all instances of **rep_item_group.**

Formal propositions:

WR1: Each instance of **rep_item_group** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'group.name' or 'group.description'.

WR2: For each instance of **rep_item_group**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.153 restrict_multi_language_for_representation

The **restrict_multi_language_for_representation** rule specifies that each instance of **representation** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'name' or 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_representation FOR
    (representation);
WHERE
  WR1:  SIZEOF( QUERY( ent <* representation |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        (amlaa.attribute_name <> 'name') AND
                        (amlaa.attribute_name <> 'description')
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* representation |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                 'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                 (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                 (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

representation: the set of all instances of representation.

Formal propositions:

WR1: Each instance of representation shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of either 'name' or 'description'.

WR2: For each instance of representation, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.154 restrict_multi_language_for_representation_item

The **restrict_multi_language_for_representation_item** rule specifies that each instance of **representation_item** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_representation_item FOR
    (representation_item);
WHERE
  WR1:  SIZEOF( QUERY( ent <* representation_item |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'name'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* representation_item |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

representation_item: the set of all instances of **representation_item.**

Formal propositions:

WR1: Each instance of **representation_item** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'name'.

WR2: For each instance of **representation_item**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.155 restrict_multi_language_for_representation_relationship

The **restrict_multi_language_for_representation_relationship** rule specifies that each instance of **representation_relationship** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_representation_relationship FOR
    (representation_relationship);
WHERE
  WR1:  SIZEOF( QUERY( ent <* representation_relationship |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* representation_relationship |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                 'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                 (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                 (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

representation_relationship: the set of all instances of **representation_relationship.**

Formal propositions:

WR1: Each instance of **representation_relationship** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **representation_relationship,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.156 restrict_multi_language_for_resource_requirement_type

The **restrict_multi_language_for_resource_requirement_type** rule specifies that each instance of **resource_requirement_type** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'name'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_resource_requirement_type FOR
    (resource_requirement_type);
WHERE
  WR1:  SIZEOF( QUERY( ent <* resource_requirement_type |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'name'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* resource_requirement_type |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                 'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                 (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                 (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

resource_requirement_type: the set of all instances of **resource_requirement_type.**

Formal propositions:

WR1: Each instance of **resource_requirement_type** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'name'.

WR2: For each instance of **resource_requirement_type**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.157 restrict_multi_language_for_retention

The **restrict_multi_language_for_retention** rule specifies that each instance of **retention** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'action.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_retention FOR
    (retention);
WHERE
  WR1:  SIZEOF( QUERY( ent <* retention |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'action.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* retention |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

retention: the set of all instances of retention.

Formal propositions:

WR1: Each instance of retention shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'action.description'.

WR2: For each instance of retention, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.158 restrict_multi_language_for_security_classification

The **restrict_multi_language_for_security_classification** rule specifies that each instance of **security_classification** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'purpose'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_security_classification FOR
    (security_classification);
WHERE
  WR1:  SIZEOF( QUERY( ent <* security_classification |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'purpose'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* security_classification |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

security_classification: the set of all instances of **security_classification.**

Formal propositions:

WR1: Each instance of **security_classification** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'purpose'.

WR2: For each instance of **security_classification**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.159 restrict_multi_language_for_shape_aspect

The **restrict_multi_language_for_shape_aspect** rule specifies that each instance of **shape_aspect** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_shape_aspect FOR
    (shape_aspect);
WHERE
  WR1:  SIZEOF( QUERY( ent <* shape_aspect |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* shape_aspect |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

shape_aspect: the set of all instances of **shape_aspect.**

Formal propositions:

WR1: Each instance of **shape_aspect** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **shape_aspect,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.160 restrict_multi_language_for_shape_aspect_relationship

The **restrict_multi_language_for_shape_aspect_relationship** rule specifies that each instance of **shape_aspect_relationship** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_shape_aspect_relationship FOR
    (shape_aspect_relationship);
WHERE
  WR1:  SIZEOF( QUERY( ent <* shape_aspect_relationship |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* shape_aspect_relationship |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

shape_aspect_relationship: the set of all instances of **shape_aspect_relationship.**

Formal propositions:

WR1: Each instance of **shape_aspect_relationship** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **shape_aspect_relationship,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.161 restrict_multi_language_for_shape_dimension_representation

The **restrict_multi_language_for_shape_dimension_representation** rule specifies that each instance of **shape_dimension_representation** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_shape_dimension_representation FOR
    (shape_dimension_representation);
WHERE
  WR1:  SIZEOF( QUERY( ent <* shape_dimension_representation |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'representation.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* shape_dimension_representation |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

shape_dimension_representation: the set of all instances of **shape_dimension_representation.**

Formal propositions:

WR1: Each instance of **shape_dimension_representation** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation.description'.

WR2: For each instance of **shape_dimension_representation,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.162 restrict_multi_language_for_shape_representation

The **restrict_multi_language_for_shape_representation** rule specifies that each instance of **shape_representation** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_shape_representation FOR
    (shape_representation);
WHERE
  WR1:  SIZEOF( QUERY( ent <* shape_representation |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'representation.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* shape_representation |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

shape_representation: the set of all instances of **shape_representation.**

Formal propositions:

WR1: Each instance of **shape_representation** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'representation.description'.

WR2: For each instance of **shape_representation**, there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.163 restrict_multi_language_for_solution_based_product_-definition

The **restrict_multi_language_for_solution_based_product_definition** rule specifies that each instance of **solution_based_product_definition** shall only be referenced by instances of **applied_multi_language_attribute_assignment** that have an attribute **attribute_name** of 'product_definition.description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_solution_based_product_definition FOR
    (solution_based_product_definition);
WHERE
  WR1:  SIZEOF( QUERY( ent <* solution_based_product_definition |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'product_definition.description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* solution_based_product_definition |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

solution_based_product_definition: the set of all instances of **solution_based_product_-definition.**

Formal propositions:

WR1: Each instance of **solution_based_product_definition** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'product_definition.description'.

WR2: For each instance of **solution_based_product_definition,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_name**.

5.2.4.164 restrict_multi_language_for_versioned_action_request

The **restrict_multi_language_for_versioned_action_request** rule specifies that each instance of **versioned_action_request** shall only be referenced by instances of **applied_multi_language_-attribute_assignment** that have an attribute **attribute_name** of 'description'.

EXPRESS specification:

```
*)
RULE restrict_multi_language_for_versioned_action_request FOR
    (versioned_action_request);
WHERE
  WR1:  SIZEOF( QUERY( ent <* versioned_action_request |
                SIZEOF( QUERY( amlaa <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                               'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        amlaa.attribute_name <> 'description'
                )) > 0
        )) = 0;
  WR2:  SIZEOF( QUERY( ent <* versioned_action_request |
                SIZEOF( QUERY( amlaa1 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                              'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                        SIZEOF( QUERY( amlaa2 <* USEDIN( ent, 'AUTOMOTIVE_DESIGN.' +
                                'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') |
                                (amlaa1.attribute_name = amlaa2.attribute_name) AND
                                (amlaa1.language = amlaa2.language)
                        )) > 1
                )) > 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

versioned_action_request: the set of all instances of **versioned_action_request.**

Formal propositions:

WR1: Each instance of **versioned_action_request** shall be only in the set of items of those instances of **multi_language_attribute_assignment** that have an attribute **attribute_name** of 'description'.

WR2: For each instance of **versioned_action_request,** there shall not be more than one **multi_language_attribute_assignment** with the same **language** and the same **attribute_-name**.

5.2.4.165 restrict_process_product_association

The **restrict_process_product_association** rule specifies that the only **name**s of **process_product_association** permitted shall be 'input', 'output', or 'produced output'. Each instance of **process_product_association** with a **name** of 'input' or 'output' shall be contained exactly once in the set of items of an instance of **process_plan_assignment** that has a role with a **name** of 'process plan context' and an assigned action that is of type **process_plan,** where the **process_plan** plays the role of the **relating_action** in an **action_relationship** with a **name** of 'process plan operation assignment' and the **related_action** is the action referenced as the process by the instance of **process_product_association** and has a **name** of 'process operation'. Each instance of **process_product_association** with a **name** of 'produced output' shall reference as process a **product_definition_process** that is a **process_plan.**

EXPRESS specification:

```
*)
RULE restrict_process_product_association FOR
    (process_product_association);
WHERE
  WR1:  SIZEOF( QUERY( ppa <* process_product_association |
                (ppa.name <> 'input') AND
                (ppa.name <> 'output') AND
                (ppa.name <> 'produced output')
        )) = 0;
  WR2:  SIZEOF( QUERY( ppa <* process_product_association |
                ((ppa.name = 'input') OR
                 (ppa.name = 'output')) AND
                (SIZEOF( QUERY( ppai <* USEDIN( ppa, 'AUTOMOTIVE_DESIGN.' +
                                        'PROCESS_PLAN_ASSIGNMENT.ITEMS') |
                         (ppai.name = 'process plan context') AND
                         ('AUTOMOTIVE_DESIGN.PROCESS_PLAN' IN
                          TYPEOF( ppai.assigned_action)) AND
                         (SIZEOF( QUERY( ar <* USEDIN( ppai.assigned_action,
                             'AUTOMOTIVE_DESIGN.ACTION_RELATIONSHIP.RELATING_ACTION') |
                                  (ar.name = 'process plan operation assignment') AND
                                  (ar.related_action :=: ppa.process) AND
                                  (ar.related_action.name = 'process operation')
                         )) = 1 )
                )) <> 1 )
        )) = 0;
  WR3:  SIZEOF( QUERY( ppa <* process_product_association |
                (ppa.name = 'produced output') AND
                (NOT ('AUTOMOTIVE_DESIGN.PROCESS_PLAN' IN
                      TYPEOF(ppa.process)))
        )) = 0;

END_RULE;
(*
```

Argument definitions:

process_product_association: the set of all instances of **process_product_association**.

Formal propositions:

WR1: For each instance of **process_product_association,** the value of the **name** attribute shall be either 'input', 'output', or 'produced output'.

WR2: If an instance of **process_product_association** has a **name** of 'input' or 'output', it shall be contained exactly once in the set of items of an instance of **process_plan_assignment** that has a role with a **name** of 'process plan context' and an assigned action that is of type **process_plan,** where the **process_plan** plays the role of the **relating_action** in an **action_relationship** with a **name** of 'process plan operation assignment' and the **related_action** is the action referenced as the process by the instance of **process_product_association** and has a **name** of 'process operation'.

WR3: For each instance of **process_product_association** with a name of 'produced output' the **product_definition_process** that is referenced as process shall be a **process_plan.**

5.2.4.166 restrict_product_categories_for_tool_part_relationship

For each instance of **product_definition_relationship** that has a **name** of 'tool part relationship', the **product_definition** used as the **relating_product_definition** shall be associated to a **product_definition_formation** referencing a product that is referenced by a **product_related_product_category** with the **name** 'tool', the **product_definition** used as the **related_product_definition** shall be associated to a **product_definition_formation** referencing a product that is referenced by a **product_related_product_category** with the **name** 'part'. This rule enforces the requirement that an item definition relationship that is a tool part relationship relates products that are tools to products that are parts.

EXPRESS specification:

```
*)
RULE restrict_product_categories_for_tool_part_relationship FOR
    (product_definition_relationship,
     product_related_product_category);
WHERE
  WR1:  SIZEOF( QUERY( pdr <* product_definition_relationship |
         (pdr.id = 'tool part relationship') AND
         ((SIZEOF( QUERY( prpc <* USEDIN(
          pdr.relating_product_definition.formation.of_product,
          'AUTOMOTIVE_DESIGN.' +
          'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
           prpc.name = 'tool'
           )) < 1) OR
          (SIZEOF( QUERY( prpc <* USEDIN(
          pdr.related_product_definition.formation.of_product,
```

```
          'AUTOMOTIVE_DESIGN.' +
          'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') |
           prpc.name = 'part'
           )) < 1))
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_definition_relationship: the set of all instances of **product_definition_relationship**.

product_related_product_category: the set of all instances of **product_related_product_category**.

Formal propositions:

WR1: For each instance of **product_definition_relationship** that has a **name** of 'tool part relationship', the **product_definition** used as the **relating_product_definition** shall be associated to a **product_definition_formation** referencing a product that is referenced by a **product_related_product_category** with the **name** 'tool', the **product_definition** used as the **related_product_definition** shall be associated to a **product_definition_formation** referencing a product that is referenced by a **product_related_product_category** with the **name** 'part'.

5.2.4.167 restrict_product_concept_features_for_product_specification

The **restrict_product_concept_features_for_product_specification** rule specifies that each instance of **configurable_item** with a **description** of 'product specification' can only have instances of **product_concept_feature_association** in its set **item_concept_feature**s that reference as their **concept** either the same instance of **product_class** that is referenced as the **item_concept** by the **configurable_item** or that reference as their **concept** instances of **product_class** that are directly or indirectly (i.e. transitively) related to the instance of **product_class** that is referenced as the **item_concept** by the **configurable_item** through one or more instances of **product_concept_relationship** with a **name** of 'class hierarchy'.

EXPRESS specification:

```
*)
RULE restrict_product_concept_features_for_product_specification FOR
    (configurable_item);
WHERE
  WR1:  SIZEOF( QUERY( ci <* configurable_item |
                (ci.description = 'product specification') AND
```

```
                (SIZEOF(ci.item_concept_feature) <>
                 (SIZEOF( QUERY( pcfa <* ci.item_concept_feature |
                          (ci.item_concept :=: pcfa.concept) ))
                 ) +
                 (SIZEOF( QUERY( pcfa <* ci.item_concept_feature |
                          (ci.item_concept :<>: pcfa.concept) AND
                          (SIZEOF( QUERY( pc <*
                                   product_class_hierarchy(pcfa.concept,[pcfa.concept]) |
                                   pc :=: ci.item_concept
                          )) > 0) ))
                 )
                )
        )) = 0;

END_RULE;
(*
```

Argument definitions:

configurable_item: the set of all instances of **configurable_item**.

Formal propositions:

WR1: The instances of **product_class** that are in the **item_concept_feature** set of a **configurable_item** with a **description** 'product specification' shall either reference the same **product_class** that is referenced by the **configurable_item** as the **item_concept** or **product_classes** that are related to that **product_class** through instances of **product_concept_relationship** with a **name** of 'class hierarchy'.

5.2.4.168 restrict_version_assignment_for_action_directive

The **restrict_version_assignment_for_action_directive** rule specifies that each instance of **action_directive** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **action_directive**.

EXPRESS specification:

```
*)
RULE restrict_version_assignment_for_action_directive FOR
    (action_directive);
WHERE
  WR1:  SIZEOF(QUERY(ent <* action_directive |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

action_directive: identifies the set of all instances of **action_directive.**

Formal propositions:

WR1: For each instance of **action_directive,** there shall be at most one instance of **version_-assignment** that contains the instance of **action_directive** in its set of items.

5.2.4.169 restrict_version_assignment_for_configurable_item

The **restrict_version_assignment_for_configurable_item** rule specifies that each instance of **configurable_item** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **configurable_item**.

EXPRESS specification:

```
*)
RULE restrict_version_assignment_for_configurable_item FOR
    (configurable_item);
WHERE
  WR1:  SIZEOF(QUERY(ent <* configurable_item |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

configurable_item: identifies the set of all instances of **configurable_item.**

Formal propositions:

WR1: For each instance of **configurable_item**, there shall be at most one instance of **version_assignment** that contains the instance of **configurable_item** in its set of items.

5.2.4.170 restrict_version_assignment_for_document

The **restrict_version_assignment_for_document** rule specifies that each instance of **document** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **document**.

EXPRESS specification:

```
*)
```

```
RULE restrict_version_assignment_for_document FOR
    (document);
WHERE
  WR1:  SIZEOF(QUERY(ent <* document |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

document: identifies the set of all instances of document.

Formal propositions:

WR1: For each instance of document, there shall be at most one instance of **version_assignment** that contains the instance of **document** in its set of items.

5.2.4.171 restrict_version_assignment_for_draughting_model

The **restrict_version_assignment_for_draughting_model** rule specifies that each instance of **draughting_model** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **draughting_model**.

EXPRESS specification:

```
*)
RULE restrict_version_assignment_for_draughting_model FOR
    (draughting_model);
WHERE
  WR1:  SIZEOF(QUERY(ent <* draughting_model |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

draughting_model: identifies the set of all instances of **draughting_model.**

Formal propositions:

WR1: For each instance of **draughting_model**, there shall be at most one instance of **version_assignment** that contains the instance of **draughting_model** in its set of items.

5.2.4.172 restrict_version_assignment_for_mechanical_design_geometric_presentation_representation

The **restrict_version_assignment_for_mechanical_design_geometric_presentation_representation** rule specifies that each instance of **mechanical_design_geometric_presentation_representation** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **mechanical_design_geometric_presentation_representation**.

EXPRESS specification:

```
*)
RULE
restrict_version_assignment_for_mechanical_design_geometric_presentation_representation
  FOR (mechanical_design_geometric_presentation_representation);
WHERE
  WR1:  SIZEOF(QUERY(ent <* mechanical_design_geometric_presentation_representation |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

mechanical_design_geometric_presentation_representation: identifies the set of all instances of **mechanical_design_geometric_presentation_representation.**

Formal propositions:

WR1: For each instance of **mechanical_design_geometric_presentation_representation**, there shall be at most one instance of **version_assignment** that contains the instance of **mechanical_design_geometric_presentation_representation** in its set of items.

5.2.4.173 restrict_version_assignment_for_physically_modelled_product_definition

The **restrict_version_assignment_for_physically_modelled_product_definition** rule specifies that each instance of **physically_modelled_product_definition** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **physically_modelled_product_definition**.

EXPRESS specification:

```
*)
RULE restrict_version_assignment_for_physically_modelled_product_definition FOR
    (physically_modelled_product_definition);
```

```
WHERE
  WR1:  SIZEOF(QUERY(ent <* physically_modelled_product_definition |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

physically_modelled_product_definition: identifies the set of all instances of **physically_modelled_product_definition.**

Formal propositions:

WR1: For each instance of **physically_modelled_product_definition,** there shall be at most one instance of **version_assignment** that contains the instance of **physically_modelled_product_definition** in its set of items.

5.2.4.174 restrict_version_assignment_for_presentation_area

The **restrict_version_assignment_for_presentation_area** rule specifies that each instance of **presentation_area** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **presentation_area**.

EXPRESS specification:

```
*)
RULE restrict_version_assignment_for_presentation_area FOR
    (presentation_area);
WHERE
  WR1:  SIZEOF(QUERY(ent <* presentation_area |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

presentation_area: identifies the set of all instances of **presentation_area.**

Formal propositions:

WR1: For each instance of **presentation_area,** there shall be at most one instance of **version_assignment** that contains the instance of **presentation_area** in its set of items.

5.2.4.175 restrict_version_assignment_for_process_plan

The **restrict_version_assignment_for_process_plan** rule specifies that each instance of **process_plan** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **process_plan**.

EXPRESS specification:

```
*)
RULE restrict_version_assignment_for_process_plan FOR
    (process_plan);
WHERE
  WR1:  SIZEOF(QUERY(ent <* process_plan |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

process_plan: identifies the set of all instances of **process_plan.**

Formal propositions:

WR1: For each instance of **process_plan**, there shall be at most one instance of **version_assignment** that contains the instance of **process_plan** in its set of items.

5.2.4.176 restrict_version_assignment_for_product_class

The **restrict_version_assignment_for_product_class** rule specifies that each instance of **product_class** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **product_class**.

EXPRESS specification:

```
*)
RULE restrict_version_assignment_for_product_class FOR
    (product_class);
WHERE
  WR1:  SIZEOF(QUERY(ent <* product_class |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_class: identifies the set of all instances of **product_class.**

Formal propositions:

WR1: For each instance of **product_class,** there shall be at most one instance of **version_assignment** that contains the instance of **product_class** in its set of items.

5.2.4.177 restrict_version_assignment_for_product_component

The **restrict_version_assignment_for_product_component** rule specifies that each instance of **product_component** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **product_component**.

EXPRESS specification:

```
*)
RULE restrict_version_assignment_for_product_component FOR
    (product_component);
WHERE
  WR1:  SIZEOF(QUERY(ent <* product_component |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_component: identifies the set of all instances of **product_component.**

Formal propositions:

WR1: For each instance of **product_component,** there shall be at most one instance of **version_assignment** that contains the instance of **product_component** in its set of items.

5.2.4.178 restrict_version_assignment_for_product_concept_feature

The **restrict_version_assignment_for_product_concept_feature** rule specifies that each instance of **product_concept_feature** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **product_concept_feature**.

EXPRESS specification:

```
*)
RULE restrict_version_assignment_for_product_concept_feature FOR
    (product_concept_feature);
WHERE
  WR1:  SIZEOF(QUERY(ent <* product_concept_feature |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_concept_feature: identifies the set of all instances of **product_concept_feature.**

Formal propositions:

WR1: For each instance of **product_concept_feature,** there shall be at most one instance of **version_assignment** that contains the instance of **product_concept_feature** in its set of items.

5.2.4.179 restrict_version_assignment_for_product_definition

The **restrict_version_assignment_for_product_definition** rule specifies that each instance of **product_definition** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **product_definition**.

EXPRESS specification:

```
*)
RULE restrict_version_assignment_for_product_definition FOR
    (product_definition);
WHERE
  WR1:  SIZEOF(QUERY(ent <* product_definition |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_definition: identifies the set of all instances of **product_definition.**

Formal propositions:

WR1: For each instance of **product_definition**, there shall be at most one instance of **version_assignment** that contains the instance of **product_definition** in its set of items.

5.2.4.180 restrict_version_assignment_for_representation

The **restrict_version_assignment_for_representation** rule specifies that each instance of **representation** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **representation**.

EXPRESS specification:

```
*)
RULE restrict_version_assignment_for_representation FOR
    (representation);
WHERE
  WR1:  SIZEOF(QUERY(ent <* representation |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

representation: identifies the set of all instances of representation.

Formal propositions:

WR1: For each instance of representation, there shall be at most one instance of **version_assignment** that contains the instance of **representation** in its set of items.

5.2.4.181 restrict_version_assignment_for_shape_representation

The **restrict_version_assignment_for_shape_representation** rule specifies that each instance of **shape_representation** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **shape_representation**.

EXPRESS specification:

```
*)
RULE restrict_version_assignment_for_shape_representation FOR
    (shape_representation);
WHERE
  WR1:  SIZEOF(QUERY(ent <* shape_representation |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
```

```
        )) = 0;

END_RULE;
(*
```

Argument definitions:

shape_representation: identifies the set of all instances of **shape_representation.**

Formal propositions:

WR1: For each instance of **shape_representation,** there shall be at most one instance of **version_assignment** that contains the instance of **shape_representation** in its set of items.

5.2.4.182 restrict_version_assignment_for_solution_based_product_definition

The **restrict_version_assignment_for_solution_based_product_definition** rule specifies that each instance of **solution_based_product_definition** shall be referenced by at most one instance of **version_assignment** that specifies the version of the **solution_based_product_definition**.

EXPRESS specification:

```
*)
RULE restrict_version_assignment_for_solution_based_product_definition FOR
    (solution_based_product_definition);
WHERE
  WR1:  SIZEOF(QUERY(ent <* solution_based_product_definition |
          SIZEOF(USEDIN(ent, 'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1
        )) = 0;

END_RULE;
(*
```

Argument definitions:

solution_based_product_definition: identifies the set of all instances of **solution_based_product_definition.**

Formal propositions:

WR1: For each instance of **solution_based_product_definition,** there shall be at most one instance of **version_assignment** that contains the instance of **solution_based_product_definition** in its set of items.

5.2.4.183 retention_requires_retention_assignment

The **retention_requires_retention_assignment** rule specifies that each instance of **retention** is referenced by at least one instance of **retention_assignment** as the **assigned_action.** This rule enforces the requirement for every instance of retention period to be applied at least once.

EXPRESS specification:

```
*)
RULE retention_requires_retention_assignment FOR
    (retention);
WHERE
  WR1:  SIZEOF( QUERY( r <* retention |
                SIZEOF( QUERY( aa <* USEDIN( r, 'AUTOMOTIVE_DESIGN.' +
                                        'ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
                        'AUTOMOTIVE_DESIGN.RETENTION_ASSIGNMENT' IN
                         TYPEOF(aa) )) = 0
             )) = 0;

END_RULE;
(*
```

Argument definitions:

retention: the set of all instances of **retention**.

Formal propositions:

WR1: Each instance of **retention** shall be referenced at least once as the **assigned_action** by an instance of **retention_assignment.**

5.2.4.184 security_classification_requires_security_classification_assignment

The **security_classification_requires_security_classification_assignment** rule specifies that each instance of **security_classification** is referenced by at least one instance of **security_classification_assignment** as the **assigned_security_classification.** This rule enforces the requirement for every **security_classification** to be applied at least once.

EXPRESS specification:

```
*)
RULE security_classification_requires_security_classification_assignment FOR
    (security_classification);
WHERE
  WR1:  SIZEOF( QUERY( sc <* security_classification |
                SIZEOF( USEDIN( sc,
```

```
                        'AUTOMOTIVE_DESIGN.SECURITY_CLASSIFICATION_ASSIGNMENT.' +
                        'ASSIGNED_SECURITY_CLASSIFICATION')
                ) = 0
        )) = 0;

END_RULE;
(*
```

Argument definitions:

security_classification: the set of all instances of **security_classification**.

Formal propositions:

WR1: Each instance of **security_classification** shall be used at least once as the **assigned_security_classification** by an instance of **security_classification_assignment.**

5.2.4.185 shape_aspect_relationship_subtype_exclusiveness

The **shape_aspect_relationship_subtype_exclusiveness** rule specifies that an instance of the subtypes of **shape_aspect_relationship** shall be one of **feature_component_relationship, shape_defining_relationship, shape_aspect_deriving_relationship,** or **dimensional_location.**

EXPRESS specification:

```
*)
RULE shape_aspect_relationship_subtype_exclusiveness FOR
    (shape_aspect_relationship);
WHERE
  WR1:  SIZEOF (QUERY (sar <* shape_aspect_relationship |
                (SIZEOF (TYPEOF (sar) *
                         ['AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP',
                          'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP',
                          'AUTOMOTIVE_DESIGN.SHAPE_ASPECT_DERIVING_RELATIONSHIP',
                          'AUTOMOTIVE_DESIGN.DIMENSIONAL_LOCATION'])
                 > 1) OR
                ((SIZEOF (TYPEOF (sar)) > 2) AND
                 (SIZEOF (TYPEOF (sar) *
                         ['AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP',
                          'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP',
                          'AUTOMOTIVE_DESIGN.SHAPE_ASPECT_DERIVING_RELATIONSHIP',
                          'AUTOMOTIVE_DESIGN.DIMENSIONAL_LOCATION'])
                  = 1))
               )) = 0;

END_RULE;
(*
```

Argument definitions:

shape_aspect_relationship: the set of all instances of **shape_aspect_relationship**.

Formal propositions:

WR1: Each instance of the subtypes of **shape_aspect_relationship** shall be one of **feature_component_relationship**, **shape_defining_relationship**, **shape_aspect_deriving_relationship**, or **dimensional_location.**

5.2.4.186 shape_aspect_subtype_exclusiveness

The **shape_aspect_subtype_exclusiveness** rule specifies that an instance of **shape_aspect** shall be at most one of **bead_end**, **boss_top**, **chamfer_offset**, **circular_closed_profile**, **closed_path_profile**, **directed_angle**, **hole_bottom**, **instanced_feature**, **joggle_termination**, **linear_profile**, **ngon_closed_profile**, **open_path_profile**, **partial_circular_profile**, **path_feature_component**, **placed_feature**, **pocket_bottom**, **rectangular_closed_profile**, **rounded_u_profile**, **slot_end**, **square_u_profile**, **taper**, **tee_profile**, **thread_occurrence**, **transition_feature**, or **vee_profile.**

EXPRESS specification:

```
*)
RULE shape_aspect_subtype_exclusiveness FOR
    (shape_aspect);
WHERE
  WR1:  SIZEOF (QUERY (sa <* shape_aspect |
                (SIZEOF (TYPEOF (sa) *
                         ['AUTOMOTIVE_DESIGN.BEAD_END',
                          'AUTOMOTIVE_DESIGN.BOSS_TOP',
                          'AUTOMOTIVE_DESIGN.CHAMFER_OFFSET',
                          'AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE',
                          'AUTOMOTIVE_DESIGN.CLOSED_PATH_PROFILE',
                          'AUTOMOTIVE_DESIGN.DATUM',
                          'AUTOMOTIVE_DESIGN.DATUM_FEATURE',
                          'AUTOMOTIVE_DESIGN.DATUM_TARGET',
                          'AUTOMOTIVE_DESIGN.DIRECTED_ANGLE',
                          'AUTOMOTIVE_DESIGN.HOLE_BOTTOM',
                          'AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
                          'AUTOMOTIVE_DESIGN.JOGGLE_TERMINATION',
                          'AUTOMOTIVE_DESIGN.LINEAR_PROFILE',
                          'AUTOMOTIVE_DESIGN.NGON_CLOSED_PROFILE',
                          'AUTOMOTIVE_DESIGN.OPEN_PATH_PROFILE',
                          'AUTOMOTIVE_DESIGN.PARTIAL_CIRCULAR_PROFILE',
                          'AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT',
                          'AUTOMOTIVE_DESIGN.PLACED_FEATURE',
                          'AUTOMOTIVE_DESIGN.POCKET_BOTTOM',
                          'AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE',
                          'AUTOMOTIVE_DESIGN.ROUNDED_U_PROFILE',
```

```
                 'AUTOMOTIVE_DESIGN.SLOT_END',
                 'AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE',
                 'AUTOMOTIVE_DESIGN.TAPER',
                 'AUTOMOTIVE_DESIGN.TEE_PROFILE',
                 'AUTOMOTIVE_DESIGN.THREAD_OCCURRENCE',
                 'AUTOMOTIVE_DESIGN.TRANSITION_FEATURE',
                 'AUTOMOTIVE_DESIGN.VEE_PROFILE'])
  > 1) OR
 ((SIZEOF (TYPEOF (sa)) > 2) AND
  (SIZEOF (TYPEOF (sa) *
         ['AUTOMOTIVE_DESIGN.BEAD_END',
                 'AUTOMOTIVE_DESIGN.BOSS_TOP',
                 'AUTOMOTIVE_DESIGN.CHAMFER_OFFSET',
                 'AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE',
                 'AUTOMOTIVE_DESIGN.CLOSED_PATH_PROFILE',
                 'AUTOMOTIVE_DESIGN.DATUM',
                 'AUTOMOTIVE_DESIGN.DATUM_FEATURE',
                 'AUTOMOTIVE_DESIGN.DATUM_TARGET',
                 'AUTOMOTIVE_DESIGN.DIRECTED_ANGLE',
                 'AUTOMOTIVE_DESIGN.HOLE_BOTTOM',
                 'AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
                 'AUTOMOTIVE_DESIGN.JOGGLE_TERMINATION',
                 'AUTOMOTIVE_DESIGN.LINEAR_PROFILE',
                 'AUTOMOTIVE_DESIGN.NGON_CLOSED_PROFILE',
                 'AUTOMOTIVE_DESIGN.OPEN_PATH_PROFILE',
                 'AUTOMOTIVE_DESIGN.PARTIAL_CIRCULAR_PROFILE',
                 'AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT',
                 'AUTOMOTIVE_DESIGN.PLACED_FEATURE',
                 'AUTOMOTIVE_DESIGN.POCKET_BOTTOM',
                 'AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE',
                 'AUTOMOTIVE_DESIGN.ROUNDED_U_PROFILE',
                 'AUTOMOTIVE_DESIGN.SLOT_END',
                 'AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE',
                 'AUTOMOTIVE_DESIGN.TAPER',
                 'AUTOMOTIVE_DESIGN.TEE_PROFILE',
                 'AUTOMOTIVE_DESIGN.THREAD_OCCURRENCE',
                 'AUTOMOTIVE_DESIGN.TRANSITION_FEATURE',
                 'AUTOMOTIVE_DESIGN.VEE_PROFILE'])
   = 1))
)) = 0;

END_RULE;
(*
```

Argument definitions:

shape_aspect: the set of all instances of **shape_aspect**.

Formal propositions:

WR1: Each instance of the subtypes of **shape_aspect** shall be at most one of **bead_end**, **boss_top**, **chamfer_offset**, **circular_closed_profile**, **closed_path_profile**, **directed_angle**, **hole_bottom**, **instanced_feature**, **joggle_termination**, **linear_profile**, **ngon_closed_profile**, **open_path_profile**, **partial_circular_profile**, **path_feature_component**, **placed_feature**, **pocket_bottom**, **rectangular_closed_profile**, **rounded_u_profile**, **slot_end**, **square_u_profile**, **taper**, **tee_profile**, **thread_occurrence**, **transition_feature**, or **vee_profile**.

5.2.4.187 shape_representation_subtype_exclusiveness

The **shape_representation_subtype_exclusiveness** rule specifies that an instance of the subtypes of a **shape_representation** shall be only one of **advanced_brep_shape_representation**, **compound_shape_representation**, **csg_shape_representation**, **edge_based_wireframe_shape_representation**, **faceted_brep_shape_representation**, **featured_shape_representation**, **geometrically_bounded_2d_wireframe_representation**, **geometrically_bounded_surface_shape_representation**, **geometrically_bounded_wireframe_shape_representation**, **hybrid_3d_shape_representation**, **manifold_surface_shape_representation**, **non_manifold_surface_shape_representation**, **point_placement_shape_representation**, **shape_dimension_representation**, or **shape_representation_with_parameters**.

EXPRESS specification:

```
*)
RULE shape_representation_subtype_exclusiveness FOR
    (shape_representation);
WHERE
  WR1:  SIZEOF (QUERY (pd <* shape_representation |
         (SIZEOF (TYPEOF (pd) *
         ['AUTOMOTIVE_DESIGN.ADVANCED_BREP_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.COMPOUND_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.CSG_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.FACETED_BREP_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.FEATURED_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.HYBRID_3D_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.MANIFOLD_SURFACE_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.NON_MANIFOLD_SURFACE_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.POINT_PLACEMENT_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.SHAPE_DIMENSION_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS'])
                > 1) OR
         ((SIZEOF (TYPEOF (pd)) > 2) AND
         (SIZEOF (TYPEOF (pd) *
         ['AUTOMOTIVE_DESIGN.ADVANCED_BREP_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.COMPOUND_SHAPE_REPRESENTATION',
```

```
          'AUTOMOTIVE_DESIGN.CSG_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.FACETED_BREP_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.FEATURED_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.HYBRID_3D_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.MANIFOLD_SURFACE_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.NON_MANIFOLD_SURFACE_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.POINT_PLACEMENT_SHAPE_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.SHAPE_DIMENSION_REPRESENTATION',
          'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS'])
                = 1))
        )) = 0;


END_RULE;
(*
```

Argument definitions:

shape_representation: the set of all instances of **shape_representation**.

Formal propositions:

WR1: Each instance of the subtypes of **shape_representation** shall be one of **advanced_brep_shape_representation**, **compound_shape_representation**, **csg_shape_representation**, **edge_based_wireframe_shape_representation**, **faceted_brep_shape_representation**, **featured_shape_representation**, **geometrically_bounded_2d_wireframe_representation**, **geometrically_bounded_surface_shape_representation**, **geometrically_bounded_wireframe_shape_representation**, **hybrid_3d_shape_representation**, **manifold_surface_shape_representation**, **non_manifold_surface_shape_representation**, **point_placement_shape_representation**, **shape_dimension_representation**, or **shape_representation_with_parameters.**

5.2.4.188 sheets_belong_to_one_drawing

The **sheets_belong_to_one_drawing** rule ensures that each **drawing_sheet_revision** belongs to exactly one **drawing_revision.**

EXPRESS specification:

```
*)
RULE sheets_belong_to_one_drawing FOR
    (drawing_sheet_revision,
     drawing_sheet_revision_usage);
WHERE
```

```
  WR1:  SIZEOF (QUERY (dsr <* drawing_sheet_revision |
                 NOT (SIZEOF (QUERY (usage <* drawing_sheet_revision_usage |
                   (dsr :=: usage.area))) = 1)
               ))=0;

END_RULE;
(*
```

Argument definitions:

drawing_sheet_revision: the set of all instances of **drawing_sheet_revision**.

drawing_sheet_revision_usage: the set of all instances of **drawing_sheet_revision_usage**.

Formal propositions:

WR1: For each instance of **drawing_sheet_revision** there shall be exactly one instance of **drawing_sheet_revision_usage** which contains the instance of **drawing_sheet_revision** as its area attribute.

5.2.4.189 styled_curve

The **styled_curve** rule ensures that a **styled_item** that gives the style for a **curve** specifies a single **curve_style.**

EXPRESS specification:

```
*)
RULE styled_curve FOR
    (styled_item);
WHERE
  WR1:  SIZEOF (QUERY (crv <* QUERY (si <* styled_item |
                 ('AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF (si.item))) |
                 NOT (SIZEOF (QUERY (sty <* crv.styles |
                   NOT ((SIZEOF (sty.styles)=1) AND
                   ('AUTOMOTIVE_DESIGN.CURVE_STYLE' IN TYPEOF (sty.styles[1]))
                 )))=0
               )))=0;

END_RULE;
(*
```

Argument definitions:

styled_item: the set of all instances of **styled_item**.

Formal propositions:

WR1: Every **styled_item** that provides the style for a **curve** shall have exactly one style that is a **curve_style.**

5.2.4.190 subtype_mandatory_camera_image_2d_or_3d

The **subtype_mandatory_camera_image_2d_or_3d** rule ensures that all **camera_image** entities are **camera_image_2d_with_scale** or **camera_image_3d_with_scale** entities.

EXPRESS specification:

```
*)
RULE subtype_mandatory_camera_image_2d_or_3d FOR
    (camera_image);
WHERE
  WR1:  SIZEOF (QUERY (ci <* camera_image |
              NOT (SIZEOF (TYPEOF (ci) *
                          ['AUTOMOTIVE_DESIGN.CAMERA_IMAGE_2D_WITH_SCALE',
                           'AUTOMOTIVE_DESIGN.CAMERA_IMAGE_3D_WITH_SCALE']) = 1
              )))=0;

END_RULE;
(*
```

Argument definitions:

camera_image: the set of all instances of **camera_image**.

Formal propositions:

WR1: Each **camera_image** shall be either a **camera_image_2d_with_scale** or a **camera_image_3d_with_scale.**

5.2.4.191 subtype_mandatory_camera_model_2d_or_3d

The **subtype_mandatory_camera_model_2d_or_3d** rule ensures that all **camera_model** entities are **camera_model_d2** or **camera_model_d3** entities.

EXPRESS specification:

```
*)
RULE subtype_mandatory_camera_model_2d_or_3d FOR
    (camera_model);
WHERE
  WR1:  SIZEOF(QUERY ( cm <* camera_model |
                 NOT (SIZEOF (TYPEOF(cm) *
```

```
                    ['AUTOMOTIVE_DESIGN.CAMERA_MODEL_D2',
                     'AUTOMOTIVE_DESIGN.CAMERA_MODEL_D3']) = 1
                )) = 0;

END_RULE;
(*
```

Argument definitions:

camera_model: the set of all instances of **camera_model**.

Formal propositions:

WR1: Each **camera_model** shall be either a **camera_model_d2** or a **camera_model_d3.**

5.2.4.192 subtype_mandatory_drawing_revision

The **subtype_mandatory_drawing_revision** rule ensures that all **drawing_revision** entities are **auto_design_drawing_revision** entities.

EXPRESS specification:

```
*)
RULE subtype_mandatory_drawing_revision FOR
    (drawing_revision);
WHERE
  WR1:  SIZEOF (QUERY (dr <* drawing_revision |
               NOT ('AUTOMOTIVE_DESIGN.AUTO_DESIGN_DRAWING_REVISION'
                       IN TYPEOF (dr))
               ))=0;

END_RULE;
(*
```

Argument definitions:

drawing_revision: the set of all instances of **drawing_revision**.

Formal propositions:

WR1: Each **drawing_revision** shall be a **auto_design_drawing_revision.**

5.2.4.193 subtype_mandatory_effectivity

The **subtype_mandatory_effectivity** rule specifies the permitted usage of the **effectivity** entity only as one of its subtypes **serial_numbered_effectivity, lot_effectivity, dated_effec-**

tivity, or **event_based_effectivity.**

EXPRESS specification:

```
*)
RULE subtype_mandatory_effectivity FOR
    (effectivity);
WHERE
  WR1:  SIZEOF (QUERY (e <* effectivity |
                ((SIZEOF( TYPEOF (e)) <= 2) AND
                 (SIZEOF( TYPEOF (e) *
                         ['AUTOMOTIVE_DESIGN.SERIAL_NUMBERED_EFFECTIVITY',
                          'AUTOMOTIVE_DESIGN.LOT_EFFECTIVITY',
                          'AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY',
                          'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY'])
                 <> 1)) OR
                ((SIZEOF( TYPEOF (e)) = 3) AND
                 (SIZEOF (TYPEOF (e) *
                         ['AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY',
                          'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY'])
                 <>  2)) OR
                (SIZEOF( TYPEOF (e)) > 3)
        )) = 0;
  WR2:  SIZEOF (QUERY (e <* effectivity |
                ((SIZEOF( TYPEOF (e)) <= 2) AND
                 (SIZEOF( TYPEOF (e) *
                         ['AUTOMOTIVE_DESIGN.SERIAL_NUMBERED_EFFECTIVITY',
                          'AUTOMOTIVE_DESIGN.LOT_EFFECTIVITY',
                          'AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY',
                          'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY'])
                 <> 1)) OR
                ((SIZEOF( TYPEOF (e)) = 3) AND
                 (SIZEOF (TYPEOF (e) *
                         ['AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY',
                          'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY'])
                 <>  2)) OR
                (SIZEOF( TYPEOF (e)) > 3)
        )) = 0;

END_RULE;
(*
```

Argument definitions:

effectivity: the set of all instances of **effectivity**.

Formal propositions:

WR1: Each instance of **effectivity** shall either be a **serial_numbered_effectivity**, **lot_effectivity**, or **dated_effectivity.**

WR2: Each instance of **effectivity** shall either be a **serial_numbered_effectivity**, **lot_effectivity**, or **event_based_effectivity.**

5.2.4.194 subtype_mandatory_pre_defined_colour

The **subtype_mandatory_pre_defined_colour** rule ensures that all **pre_defined_colour** entities are **draughting_pre_defined_colour** entities.

EXPRESS specification:

```
*)
RULE subtype_mandatory_pre_defined_colour FOR
    (pre_defined_colour);
WHERE
  WR1:  SIZEOF (QUERY (pdc <* pre_defined_colour |
               NOT ('AUTOMOTIVE_DESIGN.DRAUGHTING_PRE_DEFINED_COLOUR'
                       IN TYPEOF (pdc))))=0;

END_RULE;
(*
```

Argument definitions:

pre_defined_colour: the set of all instances of **pre_defined_colour**.

Formal propositions:

WR1: Each **pre_defined_colour** shall be a **draughting_pre_defined_colour.**

5.2.4.195 subtype_mandatory_pre_defined_curve_font

The **subtype_mandatory_pre_defined_curve_font** rule ensures that all **pre_defined_curve_-font** entities are **draughting_pre_defined_curve_font** entities.

EXPRESS specification:

```
*)
RULE subtype_mandatory_pre_defined_curve_font FOR
    (pre_defined_curve_font);
WHERE
  WR1:  SIZEOF (QUERY (pdcf <* pre_defined_curve_font |
               NOT ('AUTOMOTIVE_DESIGN.DRAUGHTING_PRE_DEFINED_CURVE_FONT'
                       IN TYPEOF (pdcf))))=0;

END_RULE;
(*
```

Argument definitions:

pre_defined_curve_font: the set of all instances of **pre_defined_curve_font**.

Formal propositions:

WR1: Each **pre_defined_curve_font** shall be a **draughting_pre_defined_curve_font.**

5.2.4.196 subtype_mandatory_pre_defined_text_font

The **subtype_mandatory_pre_defined_text_font** rule ensures that all **pre_defined_text_-font** entities are **draughting_pre_defined_text_font** entities.

EXPRESS specification:

```
*)
RULE subtype_mandatory_pre_defined_text_font FOR
    (pre_defined_text_font);
WHERE
  WR1:  SIZEOF (QUERY (pdtf <* pre_defined_text_font |
               NOT ('AUTOMOTIVE_DESIGN.DRAUGHTING_PRE_DEFINED_TEXT_FONT'
                       IN TYPEOF (pdtf))))=0;

END_RULE;
(*
```

Argument definitions:

pre_defined_text_font: the set of all instances of **pre_defined_text_font**.

Formal propositions:

WR1: Each **pre_defined_text_font** shall be a **draughting_pre_defined_text_font.**

5.2.4.197 subtype_mandatory_product_concept

The **subtype_mandatory_product_concept** rule specifies the permitted usage of the **product_concept** entity only as its subtype **product_class.**

EXPRESS specification:

```
*)
RULE subtype_mandatory_product_concept FOR
    (product_concept);
WHERE
  WR1:  SIZEOF( QUERY(pc <* product_concept |
```

```
                (SIZEOF( TYPEOF(pc) ) = 1) OR
                (SIZEOF( TYPEOF(pc) * ['AUTOMOTIVE_DESIGN.PRODUCT_CLASS'] ) = 0)
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_concept: the set of all instances of **product_concept**.

Formal propositions:

WR1: Each instance of **product_concept** shall be a **product_class.**

5.2.4.198 subtype_mandatory_product_definition_with_associated_documents

The **subtype_mandatory_product_definition_with_associated_documents** rule specifies the permitted usage of the **product_definition_with_associated_documents** entity only as its subtype **physically_modelled_product_definition.**

EXPRESS specification:

```
*)
RULE subtype_mandatory_product_definition_with_associated_documents FOR
    (product_definition_with_associated_documents);
WHERE
  WR1:  SIZEOF( QUERY(pdwad <* product_definition_with_associated_documents |
                (SIZEOF( TYPEOF(pdwad) ) = 1) OR
                (SIZEOF( TYPEOF(pdwad) *
                 ['AUTOMOTIVE_DESIGN.PHYSICALLY_MODELLED_PRODUCT_DEFINITION'] ) = 0)
        )) = 0;

END_RULE;
(*
```

Argument definitions:

product_definition_with_associated_documents: the set of all instances of **product_definition_with_associated_documents**.

Formal propositions:

WR1: Each instance of **product_definition_with_associated_documents** shall be a **physically_modelled_product_definition.**

5.2.4.199 terminator_symbol_constraint

The **terminator_symbol_constraint** rule ensures that a **defined_symbol** that is used as the definition of a **terminator_symbol** is described by a **pre_defined_terminator_symbol.**

EXPRESS specification:

```
*)
RULE terminator_symbol_constraint FOR
    (terminator_symbol);
WHERE
  WR1:  SIZEOF (QUERY (ts <* terminator_symbol |
                 (('AUTOMOTIVE_DESIGN.DEFINED_SYMBOL' IN TYPEOF(ts.item))
                 AND
                 NOT ('AUTOMOTIVE_DESIGN.PRE_DEFINED_TERMINATOR_SYMBOL' IN
                         TYPEOF (ts.item\defined_symbol.definition)))
               )) = 0;

END_RULE;
(*
```

Argument definitions:

terminator_symbol: the set of all instances of **terminator_symbol**.

Formal propositions:

WR1: Each **terminator_symbol** that has an implicit definition shall be defined by a **pre_defined_terminator_symbol.**

5.2.4.200 text_font_usage

The **text_font_usage** rule ensures that every **pre_defined_text_font** and **externally_defined_text_font** is used in the presentation of a **text_literal.**

EXPRESS specification:

```
*)
RULE text_font_usage FOR
    (externally_defined_text_font,
     pre_defined_text_font);
WHERE
  WR1:  SIZEOF (QUERY (pdtf <* pre_defined_text_font |
               NOT (SIZEOF (USEDIN (pdtf, 'AUTOMOTIVE_DESIGN.' +
                                          'TEXT_LITERAL.FONT'))>=1
               )))=0;
  WR2:  SIZEOF (QUERY (edtf <* externally_defined_text_font |
               NOT (SIZEOF (USEDIN (edtf, 'AUTOMOTIVE_DESIGN.' +
```

```
                                        'TEXT_LITERAL.FONT'))>=1
              )))=0;

END_RULE;
(*
```

Argument definitions:

externally_defined_text_font: the set of all instances of **externally_defined_text_font**.

pre_defined_text_font: the set of all instances of **pre_defined_text_font**.

Formal propositions:

WR1: Each **pre_defined_text_font** shall be the font for at least one **text_literal.**

WR2: Each **externally_defined_text_font** shall be the font for at least one **text_literal.**

5.2.4.201 versioned_action_request_requires_date_and_person_-or_organization

The **versioned_action_request_requires_date_and_person_or_organization** rule specifies that each instance of **versioned_action_request** shall be referenced by at least one instance of either **applied_date_and_person_assignment**, **applied_date_and_person_and_organization_assignment**, or **applied_date_and_organization_assignment** that has the role with a **name** of 'notified person or organization'. This rule enforces the requirement for every work request to have at least one notified person associated to it.

EXPRESS specification:

```
*)
RULE versioned_action_request_requires_date_and_person_or_organization FOR
    (versioned_action_request);
WHERE
  WR1:  SIZEOF( QUERY( v_a_r <* versioned_action_request |
                (SIZEOF( QUERY( adapa <* USEDIN( v_a_r, 'AUTOMOTIVE_DESIGN.' +
                         'APPLIED_DATE_AND_PERSON_ASSIGNMENT.ITEMS') |
                         adapa.role.name = 'notified person or organization' )) +
                 SIZEOF( QUERY( adapaoa <* USEDIN( v_a_r, 'AUTOMOTIVE_DESIGN.' +
                         'APPLIED_DATE_AND_PERSON_AND_ORGANIZATION_ASSIGNMENT.' +
                         'ITEMS') |
                         adapaoa.role.name = 'notified person or organization' )) +
                 SIZEOF( QUERY( adaoa <* USEDIN( v_a_r, 'AUTOMOTIVE_DESIGN.' +
                         'APPLIED_DATE_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
                         adaoa.role.name = 'notified person or organization' ))
                ) = 0
        )) = 0;

END_RULE;
```

```
(*
```

Argument definitions:

versioned_action_request: the set of all instances of **versioned_action_request**.

Formal propositions:

WR1: For each instance of **versioned_action_request** there shall be at least one instance of either **applied_date_and_person_assignment**, **applied_date_and_person_and_organization**, or **applied_date_and_organization** that contains the instance of **versioned_action_request** in its set of **items** and has a role with a **name** of 'notified person or organization'.

5.2.5 automotive_design function definitions

5.2.5.1 assembly_shape_is_defined

The **assembly_shape_is_defined** function accepts a **next_assembly_usage_occurrence** as input and returns a boolean result. The function will return TRUE if the shape is defined for the **product_definition** which is the **related_product_definition** and the shape is defined for the **product_definition** which is the **relating_product_definition** in the **next_assembly_usage_occurrence**, and the two shapes are related through a **shape_representation_relationship**, and the two relationships are related through **context_dependent_shape_representation**. The function will also return TRUE if the shape of the **related_product_definition** or the **relating_product_definition** is not defined.

The only time this function will return false is when the shapes are defined for the **related_product_definition** and the **relating_product_definition** and related through **shape_representation_relationship**, but the **next_assembly_usage_occurrence** and the **shape_representation_relationship** are not explicitly related through **context_dependent_shape_representation**.

NOTE – The definition of this function is identical to the definition in ISO 10303-203.

EXPRESS specification:

```
*)
FUNCTION assembly_shape_is_defined
  (assy: next_assembly_usage_occurrence) : BOOLEAN;
  LOCAL
        sdr_set : SET OF shape_definition_representation := [];
        srr_set : SET OF shape_representation_relationship := [];
  END_LOCAL;

  -- Gather all instances of shape_definition_representation where the
  -- component part has a representation defined for it.

  sdr_set := QUERY (pdr <* bag_to_set (USEDIN
```

```
            (assy.related_product_definition,
            'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION_REPRESENTATION.' +
            'DEFINITION.DEFINITION')) |
            'AUTOMOTIVE_DESIGN.SHAPE_DEFINITION_REPRESENTATION' IN
            TYPEOF (pdr)) +
            QUERY (pdr <* bag_to_set (
            USEDIN (assy.related_product_definition,
            'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION.' +
            'DEFINITION.RELATED_PRODUCT_DEFINITION')) |
            'AUTOMOTIVE_DESIGN.SHAPE_DEFINITION_REPRESENTATION' IN
            TYPEOF (pdr));

  -- If there is a representation defined for the component part

  IF SIZEOF (sdr_set) > 0 THEN

    -- For each representation of the shape of the component part gather all
    -- instances of shape_representation_relationship where the
    -- representation of component part is related to another representation.

    REPEAT i := 1 TO HIINDEX (sdr_set);
      srr_set := QUERY (rr <* bag_to_set (
               USEDIN (sdr_set[i]\
               property_definition_representation.used_representation,
               'AUTOMOTIVE_DESIGN.REPRESENTATION_RELATIONSHIP.REP_2')) |
               'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_RELATIONSHIP' IN
               TYPEOF (rr));

      -- If there is a shape_representation_relationship where the component
      -- part's shape_representation is related to another shape_-
      -- representation.

      IF SIZEOF (srr_set) > 0 THEN

        -- For each shape_representation_relationship in that set

        REPEAT j := 1 TO HIINDEX (srr_set);

          -- If the other shape_representation in the shape_representation_-
          -- relationship is the  shape_representation of the assembly
          -- product_definition in at least one instance.

          IF SIZEOF (QUERY (pdr <* bag_to_set (USEDIN
            (srr_set[j]\representation_relationship.rep_1,
            'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION_REPRESENTATION.' +
            'USED_REPRESENTATION')) |
            'AUTOMOTIVE_DESIGN.SHAPE_DEFINITION_REPRESENTATION' IN
            TYPEOF (pdr)) *
            QUERY (pdr <* bag_to_set (
            USEDIN (assy.relating_product_definition,
            'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION_REPRESENTATION.' +
```

```
            'USED_REPRESENTATION.DEFINITION')) |
            'AUTOMOTIVE_DESIGN.SHAPE_DEFINITION_REPRESENTATION'
            IN TYPEOF (pdr))) >= 1 THEN

            -- If the shape_representation_relationship and the product_-
            -- definition_relationship of each occurrence of the component
            -- and assembly relationship is not given via the
            -- context_dependent_shape_representation then return FALSE


            IF SIZEOF (QUERY (cdsr <*
              USEDIN (assy, 'AUTOMOTIVE_DESIGN.' +
              'CONTEXT_DEPENDENT_SHAPE_REPRESENTATION.' +
              'REPRESENTED_PRODUCT_RELATION') |
              NOT ((cdsr.represented_product_relation.definition :=: assy) AND
              (cdsr.representation_relation :=: srr_set[j]))) > 0
              THEN RETURN (FALSE);
            END_IF;
          END_IF;
        END_REPEAT;
      END_IF;
    END_REPEAT;
  END_IF;

  -- If the shape of the component is not specified or there are no
  -- violations then return TRUE

  RETURN (TRUE);
END_FUNCTION; -- assembly_shape_is_defined
(*
```

Argument definitions:

assy: The **assy** identifies the **next_assembly_usage_occurrence** for which the relationships are to be checked.

5.2.5.2 product_class_hierarchy

The **product_class_hierarchy** function returns the set of **product_class** instances that are related either directly or indirectly, i.e. transitively, to a given instance of **product_class** through instances of **product_concept_relationship** that have a **name** of 'hierarchy'.

EXPRESS specification:

```
*)
FUNCTION product_class_hierarchy
         (pc1 : product_class;
          pc  : SET OF product_class): SET OF product_class;
```

```
LOCAL
 i         : INTEGER;
 j         : INTEGER;
 pc_local  : SET OF product_class;
 pcr       : SET OF product_concept_relationship;
 temp      : SET OF product_class;
 new       : BOOLEAN;
END_LOCAL;

 pc_local := pc;

 pcr := USEDIN(pc1, 'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_RELATIONSHIP.' +
                    'RELATING_PRODUCT_CONCEPT');
 pcr := pcr + USEDIN(pc1, 'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_RELATIONSHIP.' +
                          'RELATED_PRODUCT_CONCEPT');
 IF SIZEOF(pcr) = 0 THEN
   RETURN(pc_local);
 END_IF;

 REPEAT i:=1 TO HIINDEX(pcr);
   IF pcr[i].name = 'class hierarchy' THEN
     new := TRUE;
     REPEAT j:=1 TO HIINDEX(pc_local);
       IF pcr[i].relating_product_concept :=: pc_local[j] THEN
          new := FALSE;
       END_IF;
     END_REPEAT;
     IF new THEN
       pc_local := pc_local + pcr[i].relating_product_concept;
       temp := product_class_hierarchy(pcr[i].relating_product_concept,pc_local);
       IF (SIZEOF(pc_local) < SIZEOF(temp)) THEN
         pc_local := temp;
       END_IF;
     END_IF;
     REPEAT j:=1 TO HIINDEX(pc_local);
       IF pcr[i].related_product_concept :=: pc_local[j] THEN
          new := FALSE;
       END_IF;
     END_REPEAT;
     IF new THEN
       pc_local := pc_local + pcr[i].related_product_concept;
       temp := product_class_hierarchy(pcr[i].related_product_concept,pc_local);
       IF (SIZEOF(pc_local) < SIZEOF(temp)) THEN
         pc_local := temp;
       END_IF;
     END_IF;
   END_IF;
 END_REPEAT;

 RETURN(pc_local);

END_FUNCTION;
```

```
(*
```

Argument definitions:

pc: The **pc** identifies the set of **product_class** instances that have been identified to be in the set of **product_class** instances directly or indirectly related to **pc1**. This is input to the function.

pc1: The **pc1** identifies the **product_class** for which the set of directly or indirectly related **product_class** instances is determined. This is input to the function.

5.2.5.3 using_representations

The function **using_representations** returns the set of **representation**s in which a **representation_item** is used.

A **representation_item** is used in a **representation** if it is:

a) referenced in the set of **items** of the **representation**; or

b) referenced by a **representation_item** used in the **representation**.

NOTE 1 – The second condition is a recursive check allowing for a **representation_item** to be used in a **representation** by being part of a tree of related **representation_item**s. The tree is rooted in an entity used in a **representation** by fulfilling the first condition.

NOTE 2 – The definition of this function is identical to the definition in ISO 10303-203.

EXPRESS specification:

```
*)
FUNCTION using_representations
  (item : representation_item)
  : SET OF representation;
  LOCAL
    results            : SET OF representation;
    result_bag         : BAG OF representation;
    intermediate_items : SET OF representation_item;
    i                  : INTEGER;
  END_LOCAL;

  -- Find the representations in which the item is used and add to the
  -- results set.

  result_bag := USEDIN(item,'AUTOMOTIVE_DESIGN.REPRESENTATION.ITEMS');

  IF SIZEOF(result_bag) > 0 THEN
    REPEAT i := 1 TO HIINDEX(result_bag);
      results := results + result_bag[i];
    END_REPEAT;
  END_IF;
```

```
  -- Find the set of representation_items in which item is used.

  intermediate_items := QUERY(z <* bag_to_set( USEDIN(item , '')) |
    'AUTOMOTIVE_DESIGN.REPRESENTATION_ITEM' IN TYPEOF(z));

  -- If the set of intermediate items is not empty;

  IF SIZEOF(intermediate_items) > 0 THEN

    -- For each element in the set, recursively add the
    -- using_representations of that element.

    REPEAT i := 1 TO HIINDEX(intermediate_items);
      results := results + using_representations(intermediate_items[i]);
    END_REPEAT;
  END_IF;

  -- Return the set of representation in which the input item is
  -- used directly and indirectly (through intervening representation_items).

  RETURN (results);
END_FUNCTION; -- using_representations
(*
```

Argument definitions:

item: the **representation_item** for which using **representation**s are determined. This is input to the function.

EXPRESS specification:

```
*)
END_SCHEMA;  -- automotive_design
(*
```

6 Conformance requirements

Conformance to this part of ISO 10303 includes satisfying the requirements stated in this part, the requirements of the implementation method(s) supported, and the relevant requirements of the normative references.

An implementation shall support at least one of the following implementation methods: ISO 10303-21, ISO 10303-22.

Requirements with respect to implementation methods-specific requirements are specified in annex C.

The Protocol Information Conformance Statement (PICS) proforma lists the options or the

combination of options that may be included in the implementation. The PICS proforma is provided in annex D.

NOTE 1 – An implementation using the method of ISO 10303–22 will make use of a specific language binding.

NOTE 2 – ISO 10303–314 defines the abstract test suite to be used in the assessment of conformance. ISO 10303–32 describes the conformance assessment process.

The conformance classes are specified using combinations of UoFs as given in tables 33, 34, 35, and 36.

NOTE 3 – Since the UoFs have interdependencies specified by the structure defined in clause 4, not every combination of UoFs is appropriate as a conformance class. There are also user requirements that restrict the allowed combinations of UoFs. Figure 66 shows the interdependencies of the UoFs.

Table 37 specifies for each conformance class the set of AIM elements that an implementation of the conformance class shall support.

NOTE 4 – Each of the following conformance classes, with the exception of CC18, makes use of a subset of the constructs specified in the AIM EXPRESS expanded listing given in annex A. The EXPRESS specifications of the conformance classes CC1 to CC17 are included in annex L in digital form.

NOTE 5 – There are totally 18 conformance classes defined for data exchange and sharing in the following application areas:

- conformance classes 1 to 5 for CAD/CAM;
- conformance classes 6 to 10 for product structure and configuration management;
- conformance classes 11 to 13 for process planning;
- conformance classes 14 and 15 for feature based design;
- conformance classes 16 and 17 for simulation and quality control;
- conformance class 18 for complete data storage and retrieval.

The conformance classes are characterized as follows:

6.1 Conformance Class for component design with 3D shape representation (CC1)

The scope of the CC1 is:

- geometry data: wireframe_model_3d, connected_surface_model, faceted_b_rep_model, b_rep_model, csg_model, and geometrically_bounded_surface_model;
- presentation data: geometric_presentation;
- product_structure data: product_management_data and element_structure.

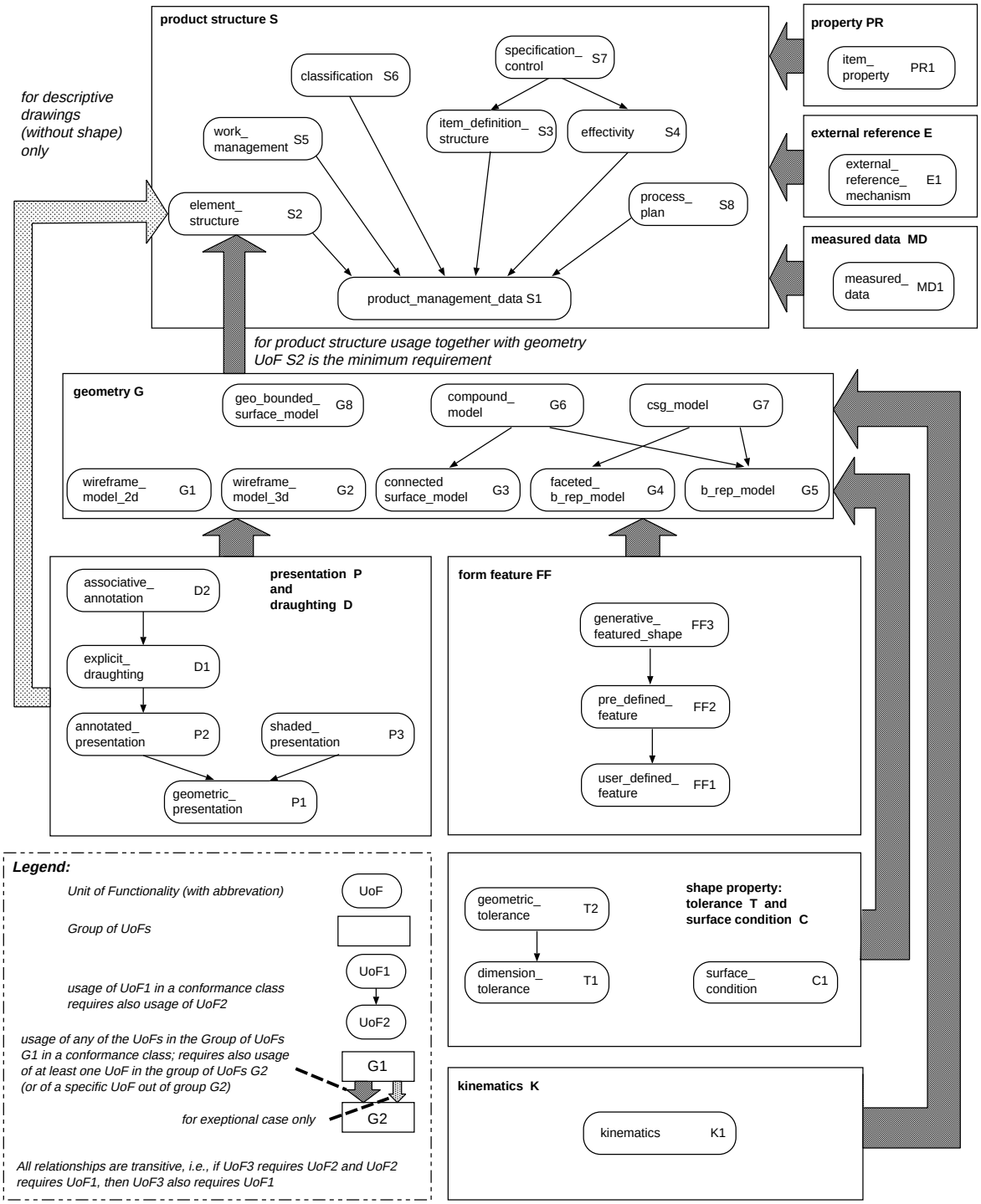


Figure 66 – Interdependencies between UoFs

NOTE 1 – This conformance class supports the following areas:

- component design of car body, power train, chassis, or interior parts;
- component design of tools.

NOTE 2 – This conformance class includes requirements that match those defined in the conformance classes 2, 4, 5, and 6 of ISO 10303–203, with the additional requirement for geometric_presentation (P1), csg_model (G7), and element_structure (S2). In the area of configuration_control information this conformance class requires product_management_data (S1), which is a subset of conformance class 1 of ISO 10303–203.

6.2 Conformance Class for assembly design with 3D shape representation (CC2)

The scope of the CC2 is:

- external_reference data;
- geometry data: wireframe_model_3d, connected_surface_model, faceted_b_rep_model, b_rep_model, csg_model, and geometrically_bounded_surface_model;
- presentation data: geometric_presentation;
- product_structure data: product_management_data, element_structure, and item_definition_structure.

NOTE 1 – This conformance class supports the following areas:

- conceptual design including assembly definitions;
- mountability examination;
- packaging layout.

NOTE 2 – This conformance class includes requirements that match those defined in the conformance classes 2, 4, 5, and 6 of ISO 10303–203, with the additional requirement for geometric_presentation (P1), csg_model (G7), element_structure (S2), and external_reference_mechanism (E1). In the area of configuration_control information this conformance class requires product_management_data (S1) and item_definition_structure (S3), which is a subset of conformance class 1 of ISO 10303–203.

NOTE 3 – This conformance class includes the requirements as defined for the conformance class 1 of this part of ISO 10303, with the additional requirement for item_definition_structure (S3) and external_reference_mechanism (E1).

6.3 Conformance Class for component drawings with wireframe or surface shape representation (CC3)

The scope of the CC3 is:

- draughting data: explicit_draughting;
- geometry data: wireframe_model_2d, wireframe_model_3d, and connected_surface_model;
- presentation data: geometric_presentation and annotated_presentation;
- product_structure data: product_management_data and element_structure.

NOTE 1 – This conformance class supports the following areas:

- component drawings or sketches for car body or some interior parts;
- component drawings or sketches for tools.

NOTE 2 – This conformance class includes requirements that match those defined in ISO 10303–201, with additional requirements for wireframe_model_3d (G2) and connected_surface_model (G3).

6.4 Conformance Class for assembly drawings with wireframe, surface or solid shape representation (CC4)

The scope of the CC4 is:

- draughting data: explicit_draughting and associative_annotation;
- external_reference data;
- geometry data: wireframe_model_2d, wireframe_model_3d, connected_surface_model, faceted_b_rep_model, b_rep_model, csg_model, and geometrically_bounded_surface_model;
- presentation data: geometric_presentation and annotated_presentation;
- product_structure data: product_management_data, element_structure, and item_definition_structure;
- tolerance data: dimension_tolerance.

NOTE 1 – This conformance class is suitable for use in the following areas:

- component or assembly drawings for power train, chassis, or interior parts;
- component or assembly drawings for tools.

NOTE 2 – This conformance class includes requirements that match those defined in the conformance classes 3, 5, 7, 9, and 10 of ISO 10303–202, with the additional requirement for item_definition_structure (S3), external_reference_mechanism (E1), and csg_model (G7).

NOTE 3 – This conformance class includes the requirements as defined for the conformance classes 2 and 3 of this part of ISO 10303.

6.5 Conformance Class for styling data (CC5)

The scope of the CC5 is:

- external_reference data;

- geometry data: connected_surface_model, faceted_b_rep_model, b_rep_model, compound_model, and csg_model;

- presentation data: geometric_presentation, annotated_presentation, and shaded_presentation;

- product_structure data: product_management_data, element_structure, and item_definition_structure.

NOTE – This conformance class is suitable for use in the following areas:

- digital mockup;

- styling.

6.6 Conformance Class for product data management without shape representation (CC6)

The scope of the CC6 is:

- external_reference data;

- property data;

- product_structure data: product_management_data, element_structure, item_definition_structure, effectivity, work_management, and classification.

NOTE 1 – This conformance class is suitable for use in the following areas:

- product data management systems that manage CAD models as files;

- administrative data of parts, assemblies, documents, and models.

NOTE 2 – This conformance class includes requirements that match those defined in the conformance class 1 of ISO 10303–203.

6.7 Conformance Class for product data management with 3D shape representation (CC7)

The scope of the CC7 is:

- external_reference data;

- geometry data: wireframe_model_3d, connected_surface_model, faceted_b_rep_model, b_rep_model, csg_model, and geometrically_bounded_surface_model;

– property data;

– presentation data: geometric_presentation;

– product_structure data: product_management_data, element_structure, item_definition_structure, effectivity, work_management, and classification.

NOTE 1 – This conformance class supports the following areas:

– administrative data of parts, assemblies, documents, and models;

– conceptual design including assembly definitions;

– mountability examination;

– packaging layout;

– data exchange between product data management systems linked to CAD/CAM systems.

NOTE 2 – This conformance class includes requirements that match those defined in the conformance classes 1, 2, 4, 5, and 6 of ISO 10303–203, with the additional requirement for geometric_presentation (P1), csg_model (G7), element_structure (S2), and external_reference_mechanism (E1).

NOTE 3 – This conformance class includes the requirements as defined for the conformance classes 2 and 6 of this part of ISO 10303.

6.8 Conformance Class for configuration controlled design without shape representation (CC8)

The scope of the CC8 is:

– external_reference data;

– property data;

– product_structure data: product_management_data, element_structure, item_definition_structure, effectivity, work_management, classification, and specification_control.

NOTE – This conformance class is suitable for use in the following areas:

– configuration control for power train, chassis, car body, or interior parts;

– configuration control for tools.

NOTE 1 – This conformance class includes the requirements as defined for the conformance class 6 of this part of ISO 10303, with the additional requirement for specification_control (S7).

6.9 Conformance Class for configuration controlled design with 3D shape representation (CC9)

The scope of the CC9 is:

– external_reference data;

– geometry data: wireframe_model_3d, connected_surface_model, faceted_b_rep_model, b_rep_model, csg_model, and geometrically_bounded_surface_model;

– property data;

– presentation data: geometric_presentation;

– product_structure data: product_management_data, element_structure, item_definition_structure, effectivity, work_management, classification, and specification_control.

NOTE 1 – This conformance class supports the following areas:

– configuration control for power train, chassis, car body, or interior parts;

– configuration control for tools;

– administrative and configuration control data of parts, assemblies, documents, and models;

– product data management systems for control of a large number of product variants.

NOTE 2 – This conformance class includes requirements that match those defined in the conformance classes 1, 2, 4, 5, and 6 of ISO 10303–203, with the additional requirement for geometric_presentation (P1), csg_model (G7), element_structure (S2), external_reference_mechanism (E1), and specification_control (S7).

NOTE 3 – This conformance class includes the requirements as defined for the conformance classes 7 and 8 of this part of ISO 10303.

6.10 Conformance Class for configuration controlled design with shape representation and draughting data (CC10)

The scope of the CC10 is:

– draughting data: explicit_draughting and associative_annotation;

– external_reference data;

– geometry data: wireframe_model_2d, wireframe_model_3d, connected_surface_model, faceted_b_rep_model, b_rep_model, csg_model, and geometrically_bounded_surface_model;

– property data;

– presentation data: geometric_presentation and annotated_presentation;

– product_structure data: product_management_data, element_structure, item_definition_structure, effectivity, work_management, classification, and specification_control;

– tolerance data: dimension_tolerance.

NOTE 1 – This conformance class supports the following areas:

– configuration control and assembly drawings for power train, chassis, car body, or interior parts;

– configuration control and assembly drawings for tools;

– administrative and configuration control data of parts, assemblies, documents, and models;

– product data management systems for control of a large number of product variants with links to CAD/CAM systems.

NOTE 2 – This conformance class includes the requirements as defined for the conformance classes 4 and 9 of this part of ISO 10303.

6.11 Conformance Class for process planning of components (CC11)

The scope of the CC11 is:

– draughting data: explicit_draughting and associative_annotation;

– external_reference data;

– geometry data: wireframe_model_2d, wireframe_model_3d, connected_surface_model, faceted_b_rep_model, b_rep_model, compound_model, and csg_model;

– measured_data;

– property data;

– presentation data: geometric_presentation and annotated_presentation;

– product_structure data: product_management_data, element_structure, classification, and process_plan.

NOTE 1 – This conformance class supports process planning for components (piece parts) with shape and draughting data.

NOTE 2 – This conformance class includes the requirements as defined for the conformance class 1 of this part of ISO 10303, without the requirement for geometrically_bounded_surface_model (G8), and the conformance class 3 of this part of ISO 10303.

6.12 Conformance Class for process planning of components with form feature and tolerance data (CC12)

The scope of the CC12 is:

– surface_condition data;

– draughting data: explicit_draughting and associative_annotation;

– external_reference data;

– form_features data: user_defined_feature, included_feature, and generative_featured_shape;

– geometry data: wireframe_model_2d, wireframe_model_3d, connected_surface_model, faceted_-b_rep_model, b_rep_model, compound_model, and csg_model;

– measured_data;

– property data;

– presentation data: geometric_presentation and annotated_presentation;

– product_structure data: product_management_data, element_structure, classification, and process_plan;

– tolerance data: dimension_tolerance and geometric_tolerance.

NOTE 1 – This conformance class supports process planning data for components (piece parts) with shape, draughting, form_feature, tolerance, and surface_condition data.

NOTE 2 – This conformance class includes the requirements as defined for the conformance class 11 of this part of ISO 10303, with the additional requirement for user_defined_feature (FF1), pre_-defined_feature (FF2), generative_featured_shape (FF3), surface_condition (C1), dimension_tolerance (T1), and geometric_tolerance (T2).

6.13 Conformance Class for effectivity controlled process planning of assemblies (CC13)

The scope of the CC13 is:

– surface_condition data;

– draughting data: explicit_draughting and associative_annotation;

– external_reference data;

– form_features data: user_defined_feature, included_feature, and generative_featured_shape;

– geometry data: wireframe_model_2d, wireframe_model_3d, connected_surface_model, faceted_-b_rep_model, b_rep_model, compound_model, and csg_model;

– measured_data;

– property data;

– presentation data: geometric_presentation and annotated_presentation;

– product_structure data: product_management_data, element_structure, item_definition_structure, effectivity, classification, and process_plan;

– tolerance data: dimension_tolerance and geometric_tolerance.

NOTE 1 – This conformance class supports process planning with effectivity control for assemblies with shape, draughting, form_feature, tolerance, and surface_condition data.

NOTE 2 – This conformance class includes the requirements as defined for the conformance class 12 of this part of ISO 10303, with the additional requirement for item_definition_structure (S3) and effectivity (S4).

6.14 Conformance Class for feature based design (CC14)

The scope of the CC14 is:

– surface_condition data;

– external_reference data;

– form_features data: user_defined_feature and included_feature;

– geometry data: wireframe_model_3d, connected_surface_model, faceted_b_rep_model, b_rep_model, and csg_model;

– property data;

– presentation data: geometric_presentation;

– product_structure data: product_management_data, element_structure, and item_definition_structure;

– tolerance data: dimension_tolerance and geometric_tolerance.

NOTE 1 – This conformance class supports the following areas:

– feature based conceptual design for components and assemblies, including manufacturing information such as tolerance and surface_condition data;

– feature based mountability examination.

NOTE 2 – This conformance class allows for identification of form features on the final shape of a component or of an assembly.

NOTE 3 – This conformance class includes requirements that match those defined in ISO 10303–224, with the additional requirement for geometric_presentation (P1), wireframe_model_3d (G2), connected_surface_model (G3), faceted_b_rep_model (G4), csg_model (G7), external_reference_mechanism (E1), and surface_condition (C1).

NOTE 4 – This conformance class includes the requirements as defined for the conformance class 2 of this part of ISO 10303, without the requirement for geometrically_bounded_surface_model (G8) and with the additional requirement for user_defined_feature (FF1), pre_defined_feature (FF2), item_property (PR1), surface_condition (C1), dimension_tolerance (T1), and geometric_tolerance (T2).

6.15 Conformance Class for feature based design with flexible feature placement (CC15)

The scope of the CC15 is:

- surface_condition data;
- external_reference data;
- form_features data: user_defined_feature, included_feature, and generative_featured_shape;
- geometry data: wireframe_model_3d, connected_surface_model, faceted_b_rep_model, b_rep_model, and csg_model;
- property data;
- presentation data: geometric_presentation;
- product_structure data: product_management_data, element_structure, and item_definition_structure;
- tolerance data: dimension_tolerance and geometric_tolerance.

NOTE 1 – This conformance class supports the following areas:

- feature based conceptual design for components and assemblies, supporting efficient design changes through flexible feature placement;
- feature based mountability examination.

NOTE 2 – This conformance class allows for an independent feature definition, e.g. in a feature library, and its usage through placement on the shape of a component or of an assembly.

NOTE 3 – This conformance class includes the requirements as defined for the conformance class 14 of this part of ISO 10303, with the additional requirement for generative_featured_shape (FF3).

6.16 Conformance Class for kinematic simulations for components and assemblies with 3d shape representation (CC16)

The scope of the CC16 is:

- external_reference data;
- geometry data: wireframe_model_3d, connected_surface_model, faceted_b_rep_model, b_rep_model, and csg_model;
- kinematics data;

- property data;
- presentation data: geometric_presentation;
- product_structure data: product_management_data, element_structure, and item_definition_structure.

NOTE 1 – This conformance class supports the following areas:

- collision detection;
- support of kinematics modules of CAD systems.

NOTE 2 – This conformance class includes the requirements as defined for the conformance class 2 of this part of ISO 10303, without the requirement for geometrically_bounded_surface_model (G8) and with the additional requirement for kinematics (K1) and item_property (PR1).

6.17 Conformance Class for measured data (CC17)

The scope of the CC17 is:

- external_reference data;
- geometry data: wireframe_model_3d;
- measured_data;
- property data;
- presentation data: geometric_presentation;
- product_structure data: product_management_data and element_structure.

NOTE – This conformance class supports the following areas:

- exchange of scanned (measured) data from a measuring system to a CAD/CAM system;
- quality control.

6.18 Conformance Class for data storage and retrieval systems (CC18)

The scope of the CC18 is:

- surface_condition data;
- draughting data: explicit_draughting and associative_annotation;
- external_reference data;

– form_features data: user_defined_feature, included_feature, and generative_featured_shape;

– geometry data: wireframe_model_2d, wireframe_model_3d, connected_surface_model, faceted_-b_rep_model, b_rep_model, compound_model, csg_model, and geometrically_bounded_surface_-model;

– kinematics data;

– measured_data;

– property data;

– presentation data: geometric_presentation, annotated_presentation, and shaded_presentation;

– product_structure data: product_management_data, element_structure, item_definition_-structure, effectivity, work_management, classification, specification_control, and process_-plan;

– tolerance data: dimension_tolerance and geometric_tolerance.

NOTE 1 – This conformance class supports database implementations to store, retrieve, or archive all of the data specified in this part of ISO 10303. Data manipulation functionality as performed in application systems is not expected to be implemented within the scope of this conformance class.

NOTE 2 – This conformance class includes all requirements as defined for the conformance classes 1 to 17 of this part of ISO 10303.

Table 33 – Usage of UoFs in Conformance Classes CC1-CC5 for CAD/CAM data exchange and sharing

	CC1	CC2	CC3	CC4	CC5
surface_condition (C1)					
explicit_draughting (D1)			•	•	
associative_annotation (D2)				•	
external_reference_mechanism (E1)		•		•	•
user_defined_feature (FF1)					
included_feature (FF2)					
generative_featured_shape (FF3)					
wireframe_model_2d (G1)			•	•	
wireframe_model_3d (G2)	•	•	•	•	
connected_surface_model (G3)	•	•	•	•	•
faceted_b_rep_model (G4)	•	•		•	•
b_rep_model (G5)	•	•		•	•
compound_model (G6)					•
csg_model (G7)	•	•		•	•
geometrically_bounded_surface_model (G8)	•	•		•	
kinematics (K1)					
measured_data (MD1)					
item_property (PR1)					
geometric_presentation (P1)	•	•	•	•	•
annotated_presentation (P2)			•	•	•
shaded_presentation (P3)					•
product_management_data (S1)	•	•	•	•	•
element_structure (S2)	•	•	•	•	•
item_definition_structure (S3)		•		•	•
effectivity (S4)					
work_management (S5)					
classification (S6)					
specification_control (S7)					
process_plan (S8)					
dimension_tolerance (T1)				•	
geometric_tolerance (T2)					

Table 34 – Usage of UoFs in Conformance Classes CC6-CC10 for product structure and configuration management data exchange and sharing

	CC6	CC7	CC8	CC9	CC10
surface_condition (C1)					
explicit_draughting (D1)					•
associative_annotation (D2)					•
external_reference_mechanism (E1)	•	•	•	•	•
user_defined_feature (FF1)					
included_feature (FF2)					
generative_featured_shape (FF3)					
wireframe_model_2d (G1)					•
wireframe_model_3d (G2)		•		•	•
connected_surface_model (G3)		•		•	•
faceted_b_rep_model (G4)		•		•	•
b_rep_model (G5)		•		•	•
compound_model (G6)					
csg_model (G7)		•		•	•
geometrically_bounded_surface_model (G8)		•		•	•
kinematics (K1)					
measured_data (MD1)					
item_property (PR1)	•	•	•	•	•
geometric_presentation (P1)		•		•	•
annotated_presentation (P2)					•
shaded_presentation (P3)					
product_management_data (S1)	•	•	•	•	•
element_structure (S2)	•	•	•	•	•
item_definition_structure (S3)	•	•	•	•	•
effectivity (S4)	•	•	•	•	•
work_management (S5)	•	•	•	•	•
classification (S6)	•	•	•	•	•
specification_control (S7)			•	•	•
process_plan (S8)					
dimension_tolerance (T1)					•
geometric_tolerance (T2)					

Table 35 – Usage of UoFs in Conformance Classes CC11-CC15 for process planning and feature based design data exchange and sharing

	CC11	CC12	CC13	CC14	CC15
surface_condition (C1)		•	•	•	•
explicit_draughting (D1)	•	•	•		
associative_annotation (D2)	•	•	•		
external_reference_mechanism (E1)	•	•	•	•	•
user_defined_feature (FF1)		•	•	•	•
included_feature (FF2)		•	•	•	•
generative_featured_shape (FF3)		•	•		•
wireframe_model_2d (G1)	•	•	•		
wireframe_model_3d (G2)	•	•	•	•	•
connected_surface_model (G3)	•	•	•	•	•
faceted_b_rep_model (G4)	•	•	•	•	•
b_rep_model (G5)	•	•	•	•	•
compound_model (G6)	•	•	•		
csg_model (G7)	•	•	•	•	•
geometrically_bounded_surface_model (G8)					
kinematics (K1)					
measured_data (MD1)	•	•	•		
item_property (PR1)	•	•	•	•	•
geometric_presentation (P1)	•	•	•	•	•
annotated_presentation (P2)	•	•	•		
shaded_presentation (P3)					
product_management_data (S1)	•	•	•	•	•
element_structure (S2)	•	•	•	•	•
item_definition_structure (S3)			•	•	•
effectivity (S4)			•		
work_management (S5)					
classification (S6)	•	•	•		
specification_control (S7)					
process_plan (S8)	•	•	•		
dimension_tolerance (T1)		•	•	•	•
geometric_tolerance (T2)		•	•	•	•

Table 36 – Usage of UoFs in Conformance Classes CC16-CC18 for simulation, quality control, and complete data storage and retrieval

	CC16	CC17	CC18
surface_condition (C1)			•
explicit_draughting (D1)			•
associative_annotation (D2)			•
external_reference_mechanism (E1)	•	•	•
user_defined_feature (FF1)			•
included_feature (FF2)			•
generative_featured_shape (FF3)			•
wireframe_model_2d (G1)			•
wireframe_model_3d (G2)	•	•	•
connected_surface_model (G3)	•		•
faceted_b_rep_model (G4)	•		•
b_rep_model (G5)	•		•
compound_model (G6)			•
csg_model (G7)	•		•
geometrically_bounded_surface_model (G8)			•
kinematics (K1)	•		•
measured_data (MD1)		•	•
item_property (PR1)	•	•	•
geometric_presentation (P1)	•	•	•
annotated_presentation (P2)			•
shaded_presentation (P3)			•
product_management_data (S1)	•	•	•
element_structure (S2)	•	•	•
item_definition_structure (S3)	•		•
effectivity (S4)			•
work_management (S5)			•
classification (S6)			•
specification_control (S7)			•
process_plan (S8)			•
dimension_tolerance (T1)			•
geometric_tolerance (T2)			•

Table 37 – Usage of AIM elements in Conformance Classes

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
action						•	•	•	•	•	•	•	•					•
action_assignment						•	•	•	•	•	•	•	•					•
action_directive						•	•	•	•	•								•
action_method						•	•	•	•	•	•	•	•					•
action_property			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
action_property_representation			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
action_relationship						•	•	•	•	•	•	•	•					•
action_request_assignment						•	•	•	•	•								•
action_request_role						•	•	•	•	•								•
action_request_solution						•	•	•	•	•								•
action_request_status						•	•	•	•	•								•
action_resource											•	•	•					•
action_resource_requirement											•	•	•					•
action_resource_type											•	•	•					•
action_role						•	•	•	•	•	•	•	•					•
action_status						•	•	•	•	•								•
address	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
advanced_brep_shape_representation	•	•		•	•		•		•	•	•	•	•	•	•	•		•
advanced_face	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
alternate_product_relationship						•	•	•	•	•								•
amount_of_substance_measure_with_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
amount_of_substance_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
angular_dimension			•	•						•	•	•	•					•
angular_location				•						•		•	•	•	•			•
angular_size				•						•		•	•	•	•			•
angularity_tolerance												•	•	•	•			•
annotation_curve_occurrence			•	•	•					•	•	•	•					•
annotation_fill_area			•	•	•					•	•	•	•					•
annotation_fill_area_occurrence			•	•	•					•	•	•	•					•
annotation_occurrence			•	•	•					•	•	•	•					•
annotation_occurrence_associativity				•						•	•	•	•					•
annotation_occurrence_relationship				•						•	•	•	•					•
annotation_subfigure_occurrence			•	•	•					•	•	•	•					•
annotation_symbol			•	•	•					•	•	•	•					•
annotation_symbol_occurrence			•	•	•					•	•	•	•					•
annotation_text			•	•	•					•	•	•	•					•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
annotation_text_occurrence			•	•	•					•	•	•	•					•
apex												•	•	•	•			•
application_context	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
application_context_element	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
application_context_element_relationship	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
application_protocol_definition	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_action_assignment						•	•	•	•	•								•
applied_action_request_assignment						•	•	•	•	•								•
applied_approval_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_certification_assignment								•	•	•								•
applied_contract_assignment						•	•	•	•	•								•
applied_date_and_organization_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_date_and_person_and_organization_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_date_and_person_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_date_and_time_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_date_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_document_reference		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_effective_context_assignment						•	•	•	•	•			•					•
applied_effective_item_assignment						•	•	•	•	•			•					•
applied_external_identification_assignment		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_group_assignment								•	•	•								•
applied_identification_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_multi_language_attribute_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_organization_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_person_and_organization_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_person_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
applied_presented_item			•	•						•	•	•	•					•
applied_security_classification_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
approval	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
approval_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
approval_date_time	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
approval_date_time_role	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
approval_person_organization	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
approval_relationship	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
approval_role	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
approval_status	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
approver_role	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
area_in_set	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
area_measure_with_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
area_presentation_relationship			•	•	•					•	•	•	•					•
area_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
assembly_component_usage		•		•	•	•	•	•	•	•	•	•	•	•	•	•		•
assembly_component_usage_substitute		•		•	•	•	•	•	•	•			•	•	•	•		•
auto_design_drawing_revision			•	•						•	•	•	•					•
axis1_placement	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
axis2_placement_2d	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
axis2_placement_3d	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
b_spline_curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
b_spline_curve_with_knots	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
b_spline_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
b_spline_surface_with_knots	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
background_colour	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
bead												•	•	•	•			•
bead_end												•	•	•	•			•
bezier_curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
bezier_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
block	•	•		•	•		•		•	•	•	•	•	•	•	•		•
boolean_result	•	•		•	•		•		•	•	•	•	•	•	•	•		•
boss												•	•	•	•			•
boss_top												•	•	•	•			•
boundary_curve	•	•		•			•		•	•								•
bounded_curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
bounded_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
box_domain	•	•		•	•		•		•	•	•	•	•	•	•	•		•
boxed_half_space	•	•		•	•		•		•	•	•	•	•	•	•	•		•
brep_with_voids	•	•		•	•		•		•	•	•	•	•	•	•	•		•
calendar_date	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
camera_image	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
camera_image_2d_with_scale	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
camera_image_3d_with_scale	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
camera_model	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
camera_model_d2	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
camera_model_d3	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
camera_model_d3_with_hlhsr					•													•
camera_usage	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
cartesian_point	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
cartesian_transformation_operator	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
cartesian_transformation_operator_2d	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
cartesian_transformation_operator_3d	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
centre_of_symmetry												•	•	•	•			•
certification								•	•	•								•
certification_assignment								•	•	•								•
certification_role								•	•	•								•
certification_type								•	•	•								•
chamfer												•	•	•	•			•
chamfer_offset												•	•	•	•			•
characterized_object						•	•	•	•	•	•	•	•	•	•			•
characterized_object_relationship												•	•	•	•			•
characterized_product_category						•	•	•	•	•	•	•	•					•
circle	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
circular_closed_profile												•	•	•	•			•
circular_pattern												•	•	•	•			•
circular_runout_tolerance												•	•	•	•			•
circularity_tolerance												•	•	•	•			•
classification_system						•	•	•	•	•	•	•	•					•
closed_path_profile												•	•	•	•			•
closed_shell	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
colour	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
colour_rgb	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
colour_specification	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
component_function_definition								•	•	•								•
composite_curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
composite_curve_on_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
composite_curve_segment	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
composite_shape_aspect												•	•	•	•			•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
composite_text			•	•	•					•	•	•	•					•
composite_text_with_associated_curves			•	•	•					•	•	•	•					•
composite_text_with_blanking_box			•	•	•					•	•	•	•					•
composite_text_with_extent			•	•	•					•	•	•	•					•
compound_feature_in_panel												•	•	•	•			•
compound_feature_in_solid												•	•	•	•			•
compound_representation_item											•	•	•				•	•
compound_shape_representation					•						•	•	•					•
concentricity_tolerance												•	•	•	•			•
concept_feature_relationship								•	•	•								•
concept_feature_relationship_with_condition								•	•	•								•
conditional_concept_feature								•	•	•								•
configurable_item								•	•	•								•
configuration_definition																•		•
configuration_design								•	•	•								•
configuration_interpolation																•		•
configuration_item								•	•	•								•
configuration_item_context	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
configuration_item_relationship								•	•	•								•
configuration_item_with_context								•	•	•								•
configured_effective_context_assignment								•	•	•								•
configured_effective_item_assignment								•	•	•								•
conic	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
conical_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
connected_edge_set					•						•	•	•					•
connected_face_set	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
connected_face_shape_representation												•	•	•	•			•
constructive_geometry_representation	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
constructive_geometry_representation_relationship	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
context_dependent_invisibility			•	•	•					•	•	•	•					•
context_dependent_over_riding_styled_item					•													•
context_dependent_shape_representation		•		•	•	•	•	•	•	•			•	•	•	•		•
context_dependent_unit	•	•				•	•	•	•	•	•	•	•	•	•	•	•	•
contract						•	•	•	•	•								•
contract_assignment						•	•	•	•	•								•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
contract_role						•	•	•	•	•								•
contract_type						•	•	•	•	•								•
conversion_based_unit	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
coordinated_universal_time_offset	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
csg_shape_representation	•	•		•	•		•		•	•	•	•	•	•	•	•		•
csg_solid	•	•		•	•		•		•	•	•	•	•	•	•	•		•
curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
curve_bounded_surface	•	•		•			•		•	•								•
curve_dimension			•	•						•	•	•	•					•
curve_replica	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
curve_style	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
curve_style_font	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
curve_style_font_pattern	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
curve_style_rendering					•													•
cylindrical_pair																•		•
cylindrical_pair_range																•		•
cylindrical_pair_value																•		•
cylindrical_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
cylindricity_tolerance												•	•	•	•			•
data_environment						•	•	•	•	•	•	•	•	•	•	•	•	•
date	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
date_and_time	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
date_and_time_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
date_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
date_role	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
date_time_role	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
dated_effectivity						•	•	•	•	•			•					•
datum												•	•	•	•			•
datum_feature												•	•	•	•			•
datum_feature_callout			•	•	•					•	•	•	•					•
datum_reference												•	•	•	•			•
datum_target												•	•	•	•			•
datum_target_callout			•	•	•					•	•	•	•					•
default_tolerance_table				•						•		•	•	•	•			•
default_tolerance_table_cell				•						•		•	•	•	•			•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
defined_symbol			•	•	•					•	•	•	•					•
definitional_representation	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
degenerate_pcurve	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
degenerate_toroidal_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
derived_shape_aspect				•						•		•	•	•	•			•
derived_unit	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
derived_unit_element	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
descriptive_representation_item		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
diameter_dimension			•	•						•	•	•	•					•
digital_document_reference		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
dimension_callout_component_relationship			•	•						•	•	•	•					•
dimension_callout_relationship			•	•						•	•	•	•					•
dimension_curve			•	•						•	•	•	•					•
dimension_curve_directed_callout			•	•						•	•	•	•					•
dimension_curve_terminator			•	•						•	•	•	•					•
dimension_pair			•	•						•	•	•	•					•
dimension_related_tolerance_zone_element												•	•	•	•			•
dimension_text_associativity				•						•	•	•	•					•
dimensional_characteristic_representation				•						•	•	•	•	•	•			•
dimensional_exponents	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
dimensional_location				•						•		•	•	•	•			•
dimensional_location_with_path				•						•		•	•	•	•			•
dimensional_size				•						•		•	•	•	•			•
directed_action						•	•	•	•	•								•
directed_angle												•	•	•	•			•
directed_dimensional_location				•						•		•	•	•	•			•
direction	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
direction_shape_representation												•	•	•	•			•
document		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
document_reference		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
document_relationship		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
document_role		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
document_type		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
document_usage_constraint		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
draughting_annotation_occurrence			•	•	•					•	•	•	•					•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
draughting_callout			•	•	•					•	•	•	•					•
draughting_callout_relationship			•	•						•	•	•	•					•
draughting_elements			•	•	•					•	•	•	•					•
draughting_model	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
draughting_pre_defined_colour	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
draughting_pre_defined_curve_font	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
draughting_pre_defined_text_font			•	•	•					•	•	•	•					•
draughting_specification_reference			•	•						•	•	•	•					•
draughting_subfigure_representation			•	•	•					•	•	•	•					•
draughting_symbol_representation			•	•	•					•	•	•	•					•
draughting_text_literal_with_delineation			•	•	•					•	•	•	•					•
draughting_title			•	•						•	•	•	•					•
drawing_definition			•	•		•	•	•	•	•	•	•	•					•
drawing_revision			•	•						•	•	•	•					•
drawing_revision_sequence			•	•						•	•	•	•					•
drawing_sheet_revision			•	•						•	•	•	•					•
drawing_sheet_revision_usage			•	•						•	•	•	•					•
edge	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
edge_based_wireframe_model					•						•	•	•					•
edge_based_wireframe_shape_representation					•						•	•	•					•
edge_curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
edge_loop	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
edge_round												•	•	•	•			•
effective_context_assignment						•	•	•	•	•			•					•
effective_item_assignment						•	•	•	•	•			•					•
effectivity						•	•	•	•	•			•					•
effectivity_assignment						•	•	•	•	•			•					•
effectivity_relationship						•	•	•	•	•			•					•
effectivity_role						•	•	•	•	•			•					•
electric_current_measure_with_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
electric_current_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
element_delivery						•	•	•	•	•								•
element_delivery_assignment						•	•	•	•	•								•
elementary_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
ellipse	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
evaluated_degenerate_pcurve	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
event_assignment						•	•	•	•	•			•					•
event_based_effectivity						•	•	•	•	•			•					•
event_context_assignment						•	•	•	•	•			•					•
event_reference						•	•	•	•	•			•					•
exclusive_product_concept_feature_category								•	•	•								•
executed_action						•	•	•	•	•								•
expression_conversion_based_unit	•	•				•	•	•	•	•	•	•	•	•	•	•	•	•
expression_definition	•	•				•	•	•	•	•	•	•	•	•	•	•	•	•
extension				•						•		•	•	•	•			•
external_identification_assignment		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
external_source	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
externally_defined_classification_attribute						•	•	•	•	•	•	•	•					•
externally_defined_classification_system						•	•	•	•	•	•	•	•					•
externally_defined_curve_font	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
externally_defined_hatch_style			•	•	•					•	•	•	•					•
externally_defined_item	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
externally_defined_product_category						•	•	•	•	•	•	•	•					•
externally_defined_representation		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
externally_defined_symbol			•	•	•					•	•	•	•					•
externally_defined_text_font			•	•	•					•	•	•	•					•
externally_defined_thread												•	•	•	•			•
externally_defined_tile_style			•	•	•					•	•	•	•					•
extruded_face_solid	•	•		•	•		•		•	•	•	•	•	•	•	•		•
face	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
face_based_surface_model					•						•	•	•					•
face_bound	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
face_outer_bound	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
face_shape_representation												•	•	•	•			•
face_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
faceted_brep	•	•		•	•		•		•	•	•	•	•	•	•	•		•
faceted_brep_shape_representation	•	•		•	•		•		•	•	•	•	•	•	•	•		•
feature_component_definition												•	•	•	•			•
feature_component_relationship												•	•	•	•			•
feature_definition												•	•	•	•			•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
feature_in_panel												•	•	•	•			•
feature_pattern												•	•	•	•			•
featured_shape_representation												•	•		•			•
fill_area_style	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
fill_area_style_colour	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
fill_area_style_hatching			•	•	•					•	•	•	•					•
fill_area_style_tile_symbol_with_style			•	•	•					•	•	•	•					•
fill_area_style_tiles			•	•	•					•	•	•	•					•
fillet												•	•	•	•			•
flatness_tolerance												•	•	•	•			•
fully_constrained_pair																•		•
functionally_defined_transformation	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
gear_pair																•		•
gear_pair_range																•		•
gear_pair_value																•		•
geometric_alignment												•	•	•	•			•
geometric_curve_set	•	•	•	•			•		•	•	•	•	•	•	•	•	•	•
geometric_intersection												•	•	•	•			•
geometric_representation_context	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
geometric_representation_item	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
geometric_set	•	•	•	•			•		•	•	•	•	•	•	•	•	•	•
geometric_tolerance												•	•	•	•			•
geometric_tolerance_relationship												•	•	•	•			•
geometric_tolerance_with_datum_reference												•	•	•	•			•
geometric_tolerance_with_defined_unit												•	•	•	•			•
geometrical_tolerance_callout			•	•	•					•	•	•	•					•
geometrically_bounded_2d_wireframe_representation			•	•						•	•	•	•					•
geometrically_bounded_surface_shape_representation	•	•		•			•		•	•								•
geometrically_bounded_wireframe_shape_representation	•	•	•	•			•		•	•	•	•	•	•	•	•	•	•
global_uncertainty_assigned_context	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
global_unit_assigned_context	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
group	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
group_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
group_member_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
group_relationship	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
group_role	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
half_space_solid	•	•		•	•		•		•	•	•	•	•	•	•	•		•
hidden_element_over_riding_styled_item					•													•
hole_bottom												•	•	•	•			•
hole_in_panel												•	•	•	•			•
hybrid_3d_shape_representation	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
hyperbola	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
identification_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
identification_role	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
initial_state																•		•
instanced_feature												•	•	•	•			•
interpolated_configuration_sequence																•		•
intersection_curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
invisibility	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
item_defined_transformation	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
joggle												•	•	•	•			•
joggle_termination												•	•	•	•			•
kinematic_analysis_result																•		•
kinematic_frame_background_representation																•		•
kinematic_frame_background_representation_association																•		•
kinematic_frame_based_transformation																•		•
kinematic_ground_representation																•		•
kinematic_joint																•		•
kinematic_link																•		•
kinematic_link_representation																•		•
kinematic_link_representation_association																•		•
kinematic_link_representation_relation																•		•
kinematic_pair																•		•
kinematic_property_definition																•		•
kinematic_property_representation_relation																•		•
kinematic_structure																•		•
known_source		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
language_role	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
leader_curve			•	•	•					•	•	•	•					•
leader_directed_callout			•	•						•	•	•	•					•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
leader_directed_dimension			•	•						•	•	•	•					•
leader_terminator			•	•	•					•	•	•	•					•
length_measure_with_unit	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
length_unit	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
light_source					•													•
light_source_ambient					•													•
light_source_directional					•													•
light_source_positional					•													•
light_source_spot					•													•
limits_and_fits				•						•		•	•	•	•			•
line	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
linear_dimension			•	•						•	•	•	•					•
linear_profile_tolerance												•	•	•	•			•
local_time	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
location_shape_representation												•	•	•	•			•
locator												•	•	•	•			•
loop	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
lot_effectivity								•	•	•								•
luminous_intensity_measure_with_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
luminous_intensity_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
make_from_usage_option											•	•	•					•
manifold_solid_brep	•	•		•	•		•		•	•	•	•	•	•	•	•		•
manifold_surface_shape_representation	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
mapped_item	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
mass_measure_with_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
mass_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
mated_item_relationship											•	•	•					•
material_designation						•	•	•	•	•	•	•	•	•	•	•	•	•
material_designation_characterization						•	•	•	•	•	•	•	•	•	•	•	•	•
material_property						•	•	•	•	•	•	•	•	•	•	•	•	•
material_property_representation						•	•	•	•	•	•	•	•	•	•	•	•	•
measure_qualification												•	•	•	•			•
measure_representation_item		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
measure_with_unit	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
mechanical_design_geometric_presentation_area	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
mechanical_design_geometric_presentation_representation	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
mechanism																•		•
mechanism_base_placement																•		•
model_change	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
modified_geometric_tolerance												•	•	•	•			•
modified_pattern												•	•	•	•			•
multi_language_attribute_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
named_unit	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
next_assembly_usage_occurrence		•		•	•	•	•	•	•	•			•	•	•	•		•
ngon_closed_profile												•	•	•	•			•
non_manifold_surface_shape_representation					•						•	•	•					•
occurrence_usage_definition								•	•	•								•
offset_curve_2d			•	•						•	•	•	•					•
offset_curve_3d	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
offset_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
one_direction_repeat_factor			•	•	•					•	•	•	•					•
open_path_profile												•	•	•	•			•
open_shell	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
ordinate_dimension			•	•						•	•	•	•					•
organization	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
organization_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
organization_relationship						•	•	•	•	•								•
organization_role	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
organizational_address	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
oriented_closed_shell	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
oriented_edge	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
oriented_face	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
oriented_open_shell	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
oriented_path	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
outer_boundary_curve	•	•		•			•		•	•								•
over_riding_styled_item	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
package_product_concept_feature								•	•	•								•
pair_actuator																•		•
pair_value																•		•
parabola	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
parallel_offset												•	•	•	•			•
parallelism_tolerance												•	•	•	•			•
parametric_representation_context	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
partial_circular_profile												•	•	•	•			•
path	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
path_feature_component												•	•	•	•			•
path_shape_representation												•	•	•	•			•
pcurve	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
perpendicular_to				•						•		•	•	•	•			•
perpendicularity_tolerance												•	•	•	•			•
person	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
person_and_organization	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
person_and_organization_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
person_and_organization_role	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
person_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
person_role	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
personal_address	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
physically_modelled_product_definition		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
placed_datum_target_feature												•	•	•	•			•
placed_feature												•	•	•	•			•
placement	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
planar_box	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
planar_curve_pair																•		•
planar_curve_pair_range																•		•
planar_extent	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
planar_pair																•		•
planar_pair_range																•		•
planar_pair_value																•		•
plane	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
plane_angle_measure_with_unit	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
plane_angle_unit	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
plus_minus_tolerance				•						•		•	•	•	•			•
pocket												•	•	•	•			•
pocket_bottom												•	•	•	•			•
point	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
point_on_curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
point_on_planar_curve_pair																•		•
point_on_planar_curve_pair_range																•		•
point_on_planar_curve_pair_value																•		•
point_on_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
point_on_surface_pair																•		•
point_on_surface_pair_range																•		•
point_on_surface_pair_value																•		•
point_placement_shape_representation											•	•	•				•	•
point_replica	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
point_style	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
poly_loop	•	•		•	•		•		•	•	•	•	•	•	•	•		•
polyline	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
position_tolerance												•	•	•	•			•
pre_defined_colour	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
pre_defined_curve_font	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
pre_defined_item	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
pre_defined_marker	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
pre_defined_point_marker_symbol	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
pre_defined_symbol	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
pre_defined_terminator_symbol	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
pre_defined_text_font			•	•	•					•	•	•	•					•
precision_qualifier				•						•		•	•	•	•			•
presentation_area	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
presentation_layer_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
presentation_layer_usage	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
presentation_representation	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
presentation_set	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
presentation_size	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
presentation_style_assignment	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
presentation_style_by_context	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
presentation_view	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
presentation_view_relationship			•	•	•					•	•	•	•					•
presented_item			•	•						•	•	•	•					•
presented_item_representation			•	•						•	•	•	•					•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
prismatic_pair																•		•
prismatic_pair_range																•		•
prismatic_pair_value																•		•
process_change			•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
process_plan											•	•	•					•
process_plan_assignment											•	•	•					•
process_product_association											•	•	•					•
process_property_association											•	•	•					•
product	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
product_category	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
product_category_relationship	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
product_class								•	•	•								•
product_component								•	•	•								•
product_concept								•	•	•								•
product_concept_context								•	•	•								•
product_concept_feature								•	•	•								•
product_concept_feature_association								•	•	•								•
product_concept_feature_category								•	•	•								•
product_concept_feature_category_usage								•	•	•								•
product_concept_relationship								•	•	•								•
product_context	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
product_definition	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
product_definition_context	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
product_definition_formation	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
product_definition_formation_relationship						•	•	•	•	•								•
product_definition_occurrence		•		•	•	•	•	•	•	•	•	•	•	•	•	•		•
product_definition_process											•	•	•					•
product_definition_relationship		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
product_definition_resource											•	•	•					•
product_definition_shape	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
product_definition_substitute		•		•	•	•	•	•	•	•			•	•	•	•		•
product_definition_usage		•		•	•	•	•	•	•	•	•	•	•	•	•	•		•
product_definition_with_associated_documents		•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
product_related_product_category	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
project						•	•	•	•	•								•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
project_assignment								•	•	•								•
projected_zone_definition												•	•	•	•			•
projection_curve			•	•						•	•	•	•					•
projection_directed_callout			•	•						•	•	•	•					•
promissory_usage_occurrence		•		•	•	•	•	•	•	•			•	•	•	•		•
property_change						•	•	•	•	•	•	•	•	•	•	•	•	•
property_definition	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
property_definition_relationship						•	•	•	•	•	•	•	•	•	•	•	•	•
property_definition_representation	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
property_process											•	•	•					•
property_value_representation						•	•	•	•	•	•	•	•	•	•	•	•	•
qualified_representation_item	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
qualitative_uncertainty	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
quantified_assembly_component_usage		•		•	•	•	•	•	•	•			•	•	•	•		•
quasi_uniform_curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
quasi_uniform_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
rack_and_pinion_pair																•		•
rack_and_pinion_pair_range																•		•
rack_and_pinion_pair_value																•		•
radius_dimension			•	•						•	•	•	•					•
ratio_measure_with_unit	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ratio_unit	•	•				•	•	•	•	•	•	•	•	•	•	•	•	•
rational_b_spline_curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
rational_b_spline_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
rectangular_closed_profile												•	•	•	•			•
rectangular_composite_surface	•	•		•			•		•	•								•
rectangular_pattern												•	•	•	•			•
rectangular_trimmed_surface	•	•		•			•		•	•						•		•
referenced_modified_datum												•	•	•	•			•
relationship_condition											•	•	•					•
rep_item_group	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
reparametrised_composite_curve_segment	•	•	•	•			•		•	•	•	•	•	•	•	•	•	•
replicate_feature												•	•	•	•			•
representation	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
representation_context	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
representation_item	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
representation_item_relationship	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
representation_map	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
representation_relationship	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
representation_relationship_with_transformation	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
requirement_for_action_resource											•	•	•					•
resource_property											•	•	•					•
resource_property_representation											•	•	•					•
resource_requirement_type											•	•	•					•
retention						•	•	•	•	•			•					•
retention_assignment						•	•	•	•	•			•					•
revolute_pair																•		•
revolute_pair_range																•		•
revolute_pair_value																•		•
revolved_face_solid	•	•		•	•		•		•	•	•	•	•	•	•	•		•
rib												•	•	•	•			•
right_angular_wedge	•	•		•	•		•		•	•	•	•	•	•	•	•		•
right_circular_cone	•	•		•	•		•		•	•	•	•	•	•	•	•		•
right_circular_cylinder	•	•		•	•		•		•	•	•	•	•	•	•	•		•
rolling_curve_pair																•		•
rolling_curve_pair_value																•		•
rolling_surface_pair																•		•
rolling_surface_pair_value																•		•
rotation_about_direction																•		•
round_hole												•	•	•	•			•
rounded_u_profile												•	•	•	•			•
runout_zone_definition												•	•	•	•			•
runout_zone_orientation												•	•	•	•			•
runout_zone_orientation_reference_direction												•	•	•	•			•
screw_pair																•		•
screw_pair_range																•		•
screw_pair_value																•		•
seam_curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
security_classification	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
security_classification_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
security_classification_level	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
security_classification_role	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
serial_numbered_effectivity								•	•	•								•
shape_aspect	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
shape_aspect_associativity				•						•	•	•	•					•
shape_aspect_deriving_relationship				•						•		•	•	•	•			•
shape_aspect_relationship	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
shape_aspect_transition	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
shape_defining_relationship												•	•	•	•			•
shape_definition_representation	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
shape_dimension_representation				•						•	•	•	•	•	•			•
shape_representation	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
shape_representation_relationship		•		•	•	•	•	•	•	•			•	•	•	•		•
shape_representation_with_parameters												•	•	•	•			•
shell_based_surface_model	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
si_unit	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
simple_pair_range																•		•
sliding_curve_pair																•		•
sliding_curve_pair_value																•		•
sliding_surface_pair																•		•
sliding_surface_pair_value																•		•
slot												•	•	•	•			•
slot_end												•	•	•	•			•
solid_angle_measure_with_unit				•		•	•	•	•	•	•	•	•	•	•	•	•	•
solid_angle_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
solid_model	•	•		•	•		•		•	•	•	•	•	•	•	•		•
solid_replica	•	•		•	•		•		•	•	•	•	•	•	•	•		•
solution_alternative_definition								•	•	•								•
solution_based_product_definition								•	•	•								•
specified_higher_usage_occurrence								•	•	•								•
sphere	•	•		•	•		•		•	•	•	•	•	•	•	•		•
spherical_pair																•		•
spherical_pair_range																•		•
spherical_pair_value																•		•
spherical_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
square_u_profile												•	•	•	•			•
standard_uncertainty				•						•		•	•	•	•			•
straightness_tolerance												•	•	•	•			•
structured_dimension_callout			•	•						•	•	•	•					•
styled_item	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
surface_condition												•	•	•	•			•
surface_condition_callout				•						•	•	•	•					•
surface_curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
surface_of_linear_extrusion	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
surface_of_revolution	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
surface_pair																•		•
surface_pair_range																•		•
surface_patch	•	•		•			•		•	•								•
surface_profile_tolerance												•	•	•	•			•
surface_rendering_properties					•													•
surface_replica	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
surface_side_style	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
surface_style_boundary	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
surface_style_control_grid	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
surface_style_fill_area	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
surface_style_parameter_line	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
surface_style_reflectance_ambient					•													•
surface_style_reflectance_ambient_diffuse					•													•
surface_style_reflectance_ambient_diffuse_specular					•													•
surface_style_rendering					•													•
surface_style_rendering_with_properties					•													•
surface_style_segmentation_curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
surface_style_silhouette	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
surface_style_transparent					•													•
surface_style_usage	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
swept_face_solid	•	•		•	•		•		•	•	•	•	•	•	•	•		•
swept_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
symbol_colour			•	•	•					•	•	•	•					•
symbol_representation			•	•	•					•	•	•	•					•

Table 37 – Usage of AIM elements in Conformance Classes (continued)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
symbol_representation_map			•	•	•					•	•	•	•					•
symbol_style			•	•	•					•	•	•	•					•
symbol_target			•	•	•					•	•	•	•					•
symmetric_shape_aspect												•	•	•	•			•
symmetry_tolerance												•	•	•	•			•
tangent												•	•	•	•			•
taper												•	•	•	•			•
tee_profile												•	•	•	•			•
terminator_symbol			•	•	•					•	•	•	•					•
text_literal			•	•	•					•	•	•	•					•
text_literal_with_associated_curves			•	•	•					•	•	•	•					•
text_literal_with_blanking_box			•	•	•					•	•	•	•					•
text_literal_with_delineation			•	•	•					•	•	•	•					•
text_literal_with_extent			•	•	•					•	•	•	•					•
text_string_representation			•	•	•					•	•	•	•					•
text_style			•	•	•					•	•	•	•					•
text_style_for_defined_font			•	•	•					•	•	•	•					•
text_style_with_box_characteristics			•	•	•					•	•	•	•					•
text_style_with_mirror			•	•	•					•	•	•	•					•
text_style_with_spacing			•	•	•					•	•	•	•					•
thermodynamic_temperature_measure_with_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
thermodynamic_temperature_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
thread												•	•	•	•			•
thread_occurrence												•	•	•	•			•
time_measure_with_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
time_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
tolerance_value				•						•		•	•	•	•			•
tolerance_zone												•	•	•	•			•
tolerance_zone_definition												•	•	•	•			•
tolerance_zone_form												•	•	•	•			•
topological_representation_item	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
toroidal_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
torus	•	•		•	•		•		•	•	•	•	•	•	•	•		•
total_runout_tolerance												•	•	•	•			•
transition_feature												•	•	•	•			•

Table 37 – Usage of AIM elements in Conformance Classes (concluded)

AIM element	Conformance class																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
trimmed_curve	•	•	•	•			•		•	•	•	•	•	•	•	•	•	•
two_direction_repeat_factor			•	•	•					•	•	•	•					•
type_qualifier				•						•		•	•	•	•			•
uncertainty_assigned_representation	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
uncertainty_measure_with_unit	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
uncertainty_qualifier	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
unconstrained_pair																•		•
unconstrained_pair_value																•		•
uniform_curve	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
uniform_surface	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
universal_pair																•		•
universal_pair_range																•		•
universal_pair_value																•		•
vector	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
vector_style	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
vee_profile												•	•	•	•			•
version_assignment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
versioned_action_request						•	•	•	•	•								•
vertex	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
vertex_loop	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
vertex_point	•	•	•	•	•		•		•	•	•	•	•	•	•	•		•
view_volume	•	•	•	•	•		•		•	•	•	•	•	•	•	•	•	•
volume_measure_with_unit						•	•	•	•	•	•	•	•	•	•	•	•	•
volume_unit						•	•	•	•	•	•	•	•	•	•	•	•	•

Annex A
(normative)
AIM EXPRESS expanded listing

This annex provides a listing of the table of short names and a listing of the EXPRESS specified in the AIM of this part of ISO 10303. No text or annotation is included. This annex is provided only in computer-interpretable form.

NOTE – The information provided on this diskette is informative; the normative text is that contained in the body of this part of ISO 10303.

EXPRESS specification:

```
*)

SCHEMA automotive_design;


 TYPE action_item = SELECT
      (action,action_method,action_property,assembly_component_usage,
       assembly_component_usage_substitute,auto_design_drawing_revision,
       conditional_concept_feature,configurable_item,configuration_design,
       configured_effective_context_assignment,
       configured_effective_item_assignment,document,draughting_model,
       kinematic_property_definition,material_designation,
       mechanical_design_geometric_presentation_representation,
       physically_modelled_product_definition,presentation_area,process_plan,
       product,product_class,product_component,product_concept_feature,
       product_concept_feature_association,product_concept_feature_category,
       product_concept_feature_category_usage,product_definition,
       product_definition_formation,product_definition_occurrence,
       product_definition_relationship,product_definition_substitute,project,
       property_definition,representation,shape_representation,
       solution_based_product_definition);
 END_TYPE;


 TYPE action_request_item = SELECT
      (action_method,action_property,assembly_component_usage,
       assembly_component_usage_substitute,auto_design_drawing_revision,
       conditional_concept_feature,configurable_item,configuration_design,
       configured_effective_context_assignment,
       configured_effective_item_assignment,document,draughting_model,
       kinematic_property_definition,material_designation,
       mechanical_design_geometric_presentation_representation,
       physically_modelled_product_definition,presentation_area,process_plan,
       product,product_class,product_component,product_concept_feature,
       product_concept_feature_association,product_concept_feature_category,
       product_concept_feature_category_usage,product_definition,
       product_definition_formation,product_definition_occurrence,
       product_definition_relationship,product_definition_substitute,
```

```
      property_definition,representation,shape_representation,
      solution_based_product_definition);
END_TYPE;

TYPE ahead_or_behind = ENUMERATION OF
     (ahead,behind);
END_TYPE;

TYPE amount_of_substance_measure = REAL;
END_TYPE;

TYPE angle_relator = ENUMERATION OF
     (equal,large,small);
END_TYPE;

TYPE approval_item = SELECT
     (action,action_directive,alternate_product_relationship,
      applied_action_assignment,assembly_component_usage,
      assembly_component_usage_substitute,auto_design_drawing_revision,
      conditional_concept_feature,configuration_design,
      configured_effective_context_assignment,
      configured_effective_item_assignment,document,draughting_model,
      kinematic_property_definition,material_designation,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product_class,product_component,product_concept_feature,
      product_concept_feature_association,product_concept_feature_category,
      product_concept_feature_category_usage,product_definition,
      product_definition_formation,product_definition_occurrence,
      product_definition_relationship,product_definition_substitute,project,
      representation,shape_representation,solution_based_product_definition);
END_TYPE;

TYPE area_measure = REAL;
END_TYPE;

TYPE area_or_view = SELECT
     (presentation_area,presentation_view);
END_TYPE;

TYPE attribute_type = SELECT
     (label,text);
END_TYPE;

TYPE axis2_placement = SELECT
     (axis2_placement_2d,axis2_placement_3d);
END_TYPE;
```

```
TYPE b_spline_curve_form = ENUMERATION OF
     (polyline_form,circular_arc,elliptic_arc,parabolic_arc,hyperbolic_arc,
      unspecified);
END_TYPE;


TYPE b_spline_surface_form = ENUMERATION OF
     (plane_surf,cylindrical_surf,conical_surf,spherical_surf,toroidal_surf,
      surf_of_revolution,ruled_surf,generalised_cone,quadric_surf,
      surf_of_linear_extrusion,unspecified);
END_TYPE;


TYPE boolean_operand = SELECT
     (solid_model,half_space_solid,csg_primitive,boolean_result);
END_TYPE;


TYPE boolean_operator = ENUMERATION OF
     (union,intersection,difference);
END_TYPE;


TYPE box_characteristic_select = SELECT
     (box_height,box_width,box_slant_angle,box_rotate_angle);
END_TYPE;


TYPE box_height = positive_ratio_measure;
END_TYPE;


TYPE box_rotate_angle = plane_angle_measure;
END_TYPE;


TYPE box_slant_angle = plane_angle_measure;
END_TYPE;


TYPE box_width = positive_ratio_measure;
END_TYPE;


TYPE category_usage_item = SELECT
     (product_class);
END_TYPE;


TYPE central_or_parallel = ENUMERATION OF
     (central,parallel);
END_TYPE;
```

```
TYPE certification_item = SELECT
     (solution_based_product_definition);
END_TYPE;


TYPE character_spacing_select = SELECT
     (length_measure,ratio_measure,measure_with_unit,descriptive_measure);
END_TYPE;


TYPE character_style_select = SELECT
     (text_style_for_defined_font);
END_TYPE;


TYPE characterized_action_definition = SELECT
     (action,action_method,action_relationship);
END_TYPE;


TYPE characterized_definition = SELECT
     (characterized_object,characterized_product_definition,shape_definition);
END_TYPE;


TYPE characterized_material_property = SELECT
     (material_property_representation);
END_TYPE;


TYPE characterized_product_definition = SELECT
     (product_definition,product_definition_relationship);
END_TYPE;


TYPE characterized_resource_definition = SELECT
     (action_resource,action_resource_requirement);
END_TYPE;


TYPE compound_item_definition = SELECT
     (list_representation_item);
END_TYPE;


TYPE configuration_design_item = SELECT
     (product_definition,product_definition_formation);
END_TYPE;


TYPE configured_effective_context_item = SELECT
     (product_concept_feature_association);
END_TYPE;
```

```
TYPE configured_effective_item = SELECT
     (action,solution_based_product_definition);
END_TYPE;


TYPE context_dependent_measure = REAL;
END_TYPE;


TYPE contract_item = SELECT
     (action,product_definition_formation);
END_TYPE;


TYPE count_measure = NUMBER;
END_TYPE;


TYPE csg_primitive = SELECT
     (sphere,block,right_angular_wedge,torus,right_circular_cone,
      right_circular_cylinder);
END_TYPE;


TYPE csg_select = SELECT
     (boolean_result,csg_primitive);
END_TYPE;


TYPE curve_font_or_scaled_curve_font_select = SELECT
     (curve_style_font_select);
END_TYPE;


TYPE curve_on_surface = SELECT
     (pcurve,surface_curve,composite_curve_on_surface);
END_TYPE;


TYPE curve_or_annotation_curve_occurrence = SELECT
     (curve,annotation_curve_occurrence);
END_TYPE;


TYPE curve_or_render = SELECT
     (curve_style,curve_style_rendering);
END_TYPE;


TYPE curve_style_font_select = SELECT
     (curve_style_font,pre_defined_curve_font,externally_defined_curve_font);
END_TYPE;
```

```
TYPE date_and_organization_item = SELECT
     (action,action_property,applied_action_assignment,
      assembly_component_usage,assembly_component_usage_substitute,
      auto_design_drawing_revision,conditional_concept_feature,
      configurable_item,configuration_design,
      configured_effective_context_assignment,
      configured_effective_item_assignment,document,draughting_model,
      kinematic_property_definition,material_designation,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product,product_class,product_component,product_concept_feature,
      product_concept_feature_association,product_concept_feature_category,
      product_concept_feature_category_usage,product_definition,
      product_definition_formation,product_definition_occurrence,
      product_definition_relationship,product_definition_substitute,project,
      property_definition,representation,shape_representation,
      solution_based_product_definition,versioned_action_request);
END_TYPE;


TYPE date_and_person_and_organization_item = SELECT
     (action,applied_action_assignment,applied_organization_assignment,
      assembly_component_usage,assembly_component_usage_substitute,
      auto_design_drawing_revision,conditional_concept_feature,
      configurable_item,configuration_design,
      configured_effective_context_assignment,
      configured_effective_item_assignment,document,draughting_model,
      kinematic_property_definition,material_designation,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product,product_class,product_component,product_concept_feature,
      product_concept_feature_association,product_concept_feature_category,
      product_concept_feature_category_usage,product_definition,
      product_definition_formation,product_definition_occurrence,
      product_definition_relationship,product_definition_substitute,project,
      property_definition,representation,shape_representation,
      solution_based_product_definition,versioned_action_request);
END_TYPE;


TYPE date_and_person_item = SELECT
     (action,action_property,applied_action_assignment,
      applied_organization_assignment,assembly_component_usage,
      assembly_component_usage_substitute,auto_design_drawing_revision,
      conditional_concept_feature,configurable_item,configuration_design,
      configured_effective_context_assignment,
      configured_effective_item_assignment,document,draughting_model,
      kinematic_property_definition,material_designation,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product,product_class,product_component,product_concept_feature,
      product_concept_feature_association,product_concept_feature_category,
      product_concept_feature_category_usage,product_definition,
```

```
      product_definition_formation,product_definition_occurrence,
      product_definition_relationship,product_definition_substitute,project,
      property_definition,representation,shape_representation,
      solution_based_product_definition,versioned_action_request);
END_TYPE;


TYPE date_and_time_item = SELECT
     (action,action_property,applied_action_assignment,
      applied_date_and_organization_assignment,
      applied_date_and_person_and_organization_assignment,
      applied_date_and_person_assignment,approval_person_organization,
      assembly_component_usage,assembly_component_usage_substitute,
      auto_design_drawing_revision,certification,conditional_concept_feature,
      configurable_item,configuration_design,
      configured_effective_context_assignment,
      configured_effective_item_assignment,document,draughting_model,
      event_based_effectivity,kinematic_property_definition,
      material_designation,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product,product_class,product_component,product_concept_feature,
      product_concept_feature_association,product_concept_feature_category,
      product_concept_feature_category_usage,product_definition,
      product_definition_formation,product_definition_occurrence,
      product_definition_relationship,product_definition_substitute,project,
      property_definition,representation,retention,shape_representation,
      solution_based_product_definition,versioned_action_request);
END_TYPE;


TYPE date_item = SELECT
     (action,action_property,applied_action_assignment,
      applied_date_and_organization_assignment,
      applied_date_and_person_and_organization_assignment,
      applied_date_and_person_assignment,approval_person_organization,
      assembly_component_usage,assembly_component_usage_substitute,
      auto_design_drawing_revision,certification,conditional_concept_feature,
      configurable_item,configuration_design,
      configured_effective_context_assignment,
      configured_effective_item_assignment,document,draughting_model,
      event_based_effectivity,kinematic_property_definition,
      material_designation,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product,product_class,product_component,product_concept_feature,
      product_concept_feature_association,product_concept_feature_category,
      product_concept_feature_category_usage,product_definition,
      product_definition_formation,product_definition_occurrence,
      product_definition_relationship,product_definition_substitute,project,
      property_definition,representation,retention,shape_representation,
      solution_based_product_definition,versioned_action_request);
END_TYPE;
```

```
TYPE date_time_select = SELECT
     (date,local_time,date_and_time);
END_TYPE;


TYPE day_in_month_number = INTEGER;
END_TYPE;


TYPE defined_symbol_select = SELECT
     (pre_defined_symbol,externally_defined_symbol);
END_TYPE;


TYPE descriptive_measure = STRING;
END_TYPE;


TYPE digital_document_reference_item = SELECT
     (externally_defined_representation);
END_TYPE;


TYPE dimension_count = INTEGER;
 WHERE
  wr1 : SELF > 0;
END_TYPE;


TYPE dimension_extent_usage = ENUMERATION OF
     (origin,target);
END_TYPE;


TYPE dimensional_characteristic = SELECT
     (dimensional_location,dimensional_size);
END_TYPE;


TYPE direction_count_select = SELECT
     (u_direction_count,v_direction_count);
END_TYPE;


TYPE document_reference_item = SELECT
     (action,action_directive,action_property,action_relationship,approval,
      assembly_component_usage,characterized_product_category,
      classification_system,configurable_item,configuration_design,
      configuration_item_relationship,contract,drawing_sheet_revision,
      exclusive_product_concept_feature_category,externally_defined_thread,
      material_designation,mechanical_design_geometric_presentation_area,
      package_product_concept_feature,presentation_area,process_plan,
      product_category,product_class,product_component,product_concept_feature
```

```
      ,product_concept_feature_category,product_definition,
      product_definition_formation_relationship,product_definition_occurrence,
      product_definition_relationship,product_definition_substitute,project,
      property_definition,representation,resource_requirement_type,retention,
      shape_aspect_relationship,solution_based_product_definition);
END_TYPE;


TYPE draughting_callout_element = SELECT
     (annotation_text_occurrence,annotation_symbol_occurrence,
      annotation_curve_occurrence);
END_TYPE;


TYPE draughting_titled_item = SELECT
     (drawing_revision,drawing_sheet_revision);
END_TYPE;


TYPE effective_context_item = SELECT
     (organization,product_definition_context);
END_TYPE;


TYPE effective_item = SELECT
     (action_property,assembly_component_usage,
      assembly_component_usage_substitute,auto_design_drawing_revision,
      conditional_concept_feature,configurable_item,configuration_design,
      document,draughting_model,kinematic_property_definition,
      material_designation,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product,product_class,product_component,product_concept_feature,
      product_concept_feature_association,product_concept_feature_category,
      product_concept_feature_category_usage,product_definition,
      product_definition_formation,product_definition_occurrence,
      product_definition_relationship,product_definition_substitute,
      property_definition,representation,security_classification,
      shape_representation,solution_based_product_definition);
END_TYPE;


TYPE electric_current_measure = REAL;
END_TYPE;


TYPE element_delivery_item = SELECT
     (applied_action_assignment);
END_TYPE;


TYPE event_context_item = SELECT
     (action,action_property,applied_action_assignment,
      assembly_component_usage,assembly_component_usage_substitute,
      auto_design_drawing_revision,conditional_concept_feature,
```

```
      configurable_item,configuration_design,
      configured_effective_context_assignment,
      configured_effective_item_assignment,document,draughting_model,
      kinematic_property_definition,material_designation,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product,product_class,product_component,product_concept_feature,
      product_concept_feature_association,product_concept_feature_category,
      product_concept_feature_category_usage,product_definition,
      product_definition_formation,product_definition_occurrence,
      product_definition_relationship,product_definition_substitute,project,
      property_definition,representation,shape_representation,
      solution_based_product_definition,versioned_action_request);
END_TYPE;


TYPE event_item = SELECT
     (event_based_effectivity,project);
END_TYPE;


TYPE expression_operand = SELECT
     (measure_with_unit,expression_definition);
END_TYPE;


TYPE external_identification_item = SELECT
     (document);
END_TYPE;


TYPE fill_area_style_tile_shape_select = SELECT
     (fill_area_style_tile_symbol_with_style);
END_TYPE;


TYPE fill_style_select = SELECT
     (fill_area_style_colour,externally_defined_tile_style,
      fill_area_style_tiles,externally_defined_hatch_style,
      fill_area_style_hatching);
END_TYPE;


TYPE font_select = SELECT
     (pre_defined_text_font,externally_defined_text_font);
END_TYPE;


TYPE geometric_set_select = SELECT
     (point,curve,surface);
END_TYPE;


TYPE group_item = SELECT
     (product_component);
```

```
END_TYPE;

TYPE group_member_item = SELECT
     (approval_status,characterized_product_category,document_type,
      drawing_definition,product_concept_feature,
      product_concept_feature_category_usage,product_definition,
      representation_item,security_classification_level,shape_aspect);
END_TYPE;

TYPE hiding_or_blanking_select = SELECT
     (presentation_area,presentation_view,annotation_fill_area);
END_TYPE;

TYPE hour_in_day = INTEGER;
 WHERE
  wr1 : {0<=SELF<=24};
END_TYPE;

TYPE identification_item = SELECT
     (approval_status,assembly_component_usage,classification_system,document,
      document_type,draughting_model,externally_defined_classification_system,
      measure_representation_item,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,product_class,product_component,
      product_concept_feature,product_concept_feature_category,
      product_definition,product_definition_occurrence,
      security_classification_level,shape_aspect_relationship,
      shape_representation,solution_based_product_definition,
      version_assignment);
END_TYPE;

TYPE identifier = STRING;
END_TYPE;

TYPE interpolation_type = ENUMERATION OF
     (undefined,synchronous,linear);
END_TYPE;

TYPE invisibility_context = SELECT
     (presentation_layer_usage,presentation_representation,presentation_set);
END_TYPE;

TYPE invisible_item = SELECT
     (styled_item,presentation_layer_assignment,presentation_representation);
END_TYPE;
```

```
TYPE kinematic_analysis_definition = SELECT
     (interpolated_configuration_sequence);
END_TYPE;


TYPE kinematic_frame_background = SELECT
     (point,curve,surface);
END_TYPE;


TYPE kinematic_result = SELECT
     (interpolated_configuration_sequence);
END_TYPE;


TYPE knot_type = ENUMERATION OF
     (uniform_knots,unspecified,quasi_uniform_knots,piecewise_bezier_knots);
END_TYPE;


TYPE label = STRING;
END_TYPE;


TYPE layered_item = SELECT
     (presentation_representation,representation_item);
END_TYPE;


TYPE length_measure = REAL;
END_TYPE;


TYPE limit_condition = ENUMERATION OF
     (maximum_material_condition,least_material_condition,
      regardless_of_feature_size);
END_TYPE;


TYPE list_of_reversible_topology_item = LIST[0:?] OF reversible_topology_item;
END_TYPE;


TYPE list_representation_item = LIST[1:?] OF representation_item;
END_TYPE;


TYPE luminous_intensity_measure = REAL;
END_TYPE;


TYPE marker_select = SELECT
     (marker_type,pre_defined_marker);
END_TYPE;
```

```
TYPE marker_type = ENUMERATION OF
     (dot,x,plus,asterisk,ring,square,triangle);
END_TYPE;


TYPE mass_measure = REAL;
END_TYPE;


TYPE measure_value = SELECT
     (length_measure,mass_measure,time_measure,electric_current_measure,
      thermodynamic_temperature_measure,amount_of_substance_measure,
      luminous_intensity_measure,plane_angle_measure,solid_angle_measure,
      area_measure,volume_measure,ratio_measure,parameter_value,
      numeric_measure,context_dependent_measure,descriptive_measure,
      positive_length_measure,positive_plane_angle_measure,
      positive_ratio_measure,count_measure);
END_TYPE;


TYPE minute_in_hour = INTEGER;
 WHERE
  wr1 : {0<=SELF<=59};
END_TYPE;


TYPE month_in_year_number = INTEGER;
 WHERE
  wr1 : {1<=SELF<=12};
END_TYPE;


TYPE motion_parameter_measure = SELECT
     (parameter_value,measure_with_unit);
END_TYPE;


TYPE multi_language_attribute_item = SELECT
     (action,action_directive,action_method,action_property,
      action_relationship,action_resource_requirement,
      alternate_product_relationship,application_context,approval,
      area_presentation_relationship,assembly_component_usage,
      assembly_component_usage_substitute,camera_image_2d_with_scale,
      camera_image_3d_with_scale,certification,characterized_product_category,
      classification_system,configurable_item,configuration_design,
      configuration_item_relationship,
      constructive_geometry_representation_relationship,data_environment,
      default_tolerance_table_cell,descriptive_representation_item,
      dimensional_location,dimensional_size,document,document_relationship,
      draughting_model,draughting_title,drawing_sheet_revision,effectivity,
      effectivity_relationship,externally_defined_classification_attribute,
      externally_defined_classification_system,
      externally_defined_product_category,externally_defined_representation,
      geometric_tolerance,group,group_relationship,identification_role,
```

```
      make_from_usage_option,material_property,measure_representation_item,
      mechanical_design_geometric_presentation_representation,model_change,
      organization_relationship,organization_role,pair_actuator,
      presentation_area,presentation_layer_assignment,presentation_view,
      presentation_view_relationship,process_change,process_plan,
      process_product_association,product,product_category,product_class,
      product_component,product_concept_feature,
      product_concept_feature_association,product_concept_feature_category,
      product_definition,product_definition_formation,
      product_definition_formation_relationship,product_definition_occurrence,
      product_definition_relationship,product_definition_relationship,
      product_definition_shape,product_definition_substitute,project,
      property_change,property_definition,property_definition_relationship,
      property_definition_representation,rep_item_group,representation,
      representation_item,representation_relationship,
      resource_requirement_type,retention,security_classification,shape_aspect
      ,shape_aspect_relationship,shape_dimension_representation,
      shape_representation,solution_based_product_definition,
      versioned_action_request);
END_TYPE;


TYPE null_style = ENUMERATION OF
     (null);
END_TYPE;


TYPE numeric_measure = NUMBER;
END_TYPE;


TYPE organization_item = SELECT
     (action,action_property,applied_action_assignment,
      applied_identification_assignment,assembly_component_usage,
      assembly_component_usage_substitute,auto_design_drawing_revision,
      conditional_concept_feature,configurable_item,configuration_design,
      configured_effective_context_assignment,
      configured_effective_item_assignment,contract,document,draughting_model,
      element_delivery,kinematic_property_definition,material_designation,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product,product_class,product_component,product_concept_feature,
      product_concept_feature_association,product_concept_feature_category,
      product_concept_feature_category_usage,product_definition,
      product_definition_formation,product_definition_occurrence,
      product_definition_relationship,product_definition_substitute,project,
      property_definition,representation,shape_representation,
      solution_based_product_definition,versioned_action_request);
END_TYPE;


TYPE parameter_value = REAL;
END_TYPE;
```

```
TYPE pcurve_or_surface = SELECT
     (pcurve,surface);
END_TYPE;


TYPE person_and_organization_item = SELECT
     (action,action_property,applied_action_assignment,
      assembly_component_usage,assembly_component_usage_substitute,
      auto_design_drawing_revision,conditional_concept_feature,
      configurable_item,configuration_design,
      configured_effective_context_assignment,
      configured_effective_item_assignment,document,draughting_model,
      kinematic_property_definition,material_designation,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product,product_class,product_component,product_concept_feature,
      product_concept_feature_association,product_concept_feature_category,
      product_concept_feature_category_usage,product_definition,
      product_definition_formation,product_definition_occurrence,
      product_definition_relationship,product_definition_substitute,project,
      property_definition,representation,shape_representation,
      solution_based_product_definition,versioned_action_request);
END_TYPE;


TYPE person_item = SELECT
     (action,action_property,applied_action_assignment,
      assembly_component_usage,assembly_component_usage_substitute,
      auto_design_drawing_revision,conditional_concept_feature,
      configurable_item,configuration_design,
      configured_effective_context_assignment,
      configured_effective_item_assignment,document,draughting_model,
      kinematic_property_definition,material_designation,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product,product_class,product_component,product_concept_feature,
      product_concept_feature_association,product_concept_feature_category,
      product_concept_feature_category_usage,product_definition,
      product_definition_formation,product_definition_occurrence,
      product_definition_relationship,product_definition_substitute,project,
      property_definition,representation,shape_representation,
      solution_based_product_definition,versioned_action_request);
END_TYPE;


TYPE person_organization_select = SELECT
     (person,organization,person_and_organization);
END_TYPE;


TYPE plane_angle_measure = REAL;
END_TYPE;
```

```
TYPE positive_length_measure = length_measure;
 WHERE
  wr1 : SELF > 0;
END_TYPE;


TYPE positive_plane_angle_measure = plane_angle_measure;
 WHERE
  wr1 : SELF > 0;
END_TYPE;


TYPE positive_ratio_measure = ratio_measure;
 WHERE
  wr1 : SELF > 0;
END_TYPE;


TYPE preferred_surface_curve_representation = ENUMERATION OF
     (curve_3d,pcurve_s1,pcurve_s2);
END_TYPE;


TYPE presentable_text = STRING;
END_TYPE;


TYPE presentation_representation_select = SELECT
     (presentation_representation,presentation_set);
END_TYPE;


TYPE presentation_size_assignment_select = SELECT
     (presentation_view,presentation_area,area_in_set);
END_TYPE;


TYPE presentation_style_select = SELECT
     (point_style,curve_style,surface_style_usage,symbol_style,fill_area_style
      ,text_style,null_style);
END_TYPE;


TYPE presented_item_select = SELECT
     (action,assembly_component_usage,configuration_design,
      exclusive_product_concept_feature_category,
      package_product_concept_feature,process_plan,product_class,
      product_component,product_concept_feature,
      product_concept_feature_category,product_definition,
      product_definition_occurrence,product_definition_relationship,
      product_definition_substitute,representation,
      solution_based_product_definition);
END_TYPE;
```

```
TYPE process_or_process_relationship = SELECT
     (product_definition_process,property_process,relationship_with_condition)
      ;
END_TYPE;


TYPE process_plan_item = SELECT
     (process_product_association);
END_TYPE;


TYPE project_item = SELECT
     (product_class);
END_TYPE;


TYPE property_or_shape_select = SELECT
     (property_definition,shape_definition);
END_TYPE;


TYPE ratio_measure = REAL;
END_TYPE;


TYPE relationship_with_condition = SELECT
     (action_relationship);
END_TYPE;


TYPE rendering_properties_select = SELECT
     (surface_style_reflectance_ambient,surface_style_transparent);
END_TYPE;


TYPE represented_definition = SELECT
     (property_definition,shape_aspect,shape_aspect_relationship);
END_TYPE;


TYPE retention_item = SELECT
     (action,action_property,applied_action_assignment,
      assembly_component_usage,assembly_component_usage_substitute,
      auto_design_drawing_revision,conditional_concept_feature,
      configurable_item,configuration_design,
      configured_effective_context_assignment,
      configured_effective_item_assignment,document,draughting_model,
      kinematic_property_definition,material_designation,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product,product_class,product_component,product_concept_feature,
      product_concept_feature_association,product_concept_feature_category,
      product_concept_feature_category_usage,product_definition,
      product_definition_formation,product_definition_occurrence,
      product_definition_relationship,product_definition_substitute,project,
```

```
      property_definition,representation,shape_representation,
      solution_based_product_definition,versioned_action_request);
END_TYPE;


TYPE reversible_topology = SELECT
     (reversible_topology_item,list_of_reversible_topology_item,
      set_of_reversible_topology_item);
END_TYPE;


TYPE reversible_topology_item = SELECT
     (edge,path,face,face_bound,closed_shell,open_shell);
END_TYPE;


TYPE rigid_placement = SELECT
     (axis2_placement_3d);
END_TYPE;


TYPE rotational_range_measure = SELECT
     (plane_angle_measure,unlimited_range);
END_TYPE;


TYPE second_in_minute = REAL;
 WHERE
  wr1 : {0<=SELF<=60};
END_TYPE;


TYPE security_classification_item = SELECT
     (action,action_directive,action_property,applied_action_assignment,
      assembly_component_usage,assembly_component_usage_substitute,
      auto_design_drawing_revision,configuration_design,
      configured_effective_context_assignment,
      configured_effective_item_assignment,document,draughting_model,
      kinematic_property_definition,material_designation,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product,product_class,product_component,product_concept_feature,
      product_concept_feature_category,product_definition,
      product_definition_formation,project,property_definition,representation,
      shape_representation,solution_based_product_definition,
      versioned_action_request);
END_TYPE;


TYPE set_of_reversible_topology_item = SET[0:?] OF reversible_topology_item;
END_TYPE;


TYPE shading_curve_method = ENUMERATION OF
     (constant_colour,linear_colour);
```

```
END_TYPE;

TYPE shading_surface_method = ENUMERATION OF
     (constant_shading,colour_shading,dot_shading,normal_shading);
END_TYPE;

TYPE shape_definition = SELECT
     (product_definition_shape,shape_aspect,shape_aspect_relationship);
END_TYPE;

TYPE shell = SELECT
     (open_shell,closed_shell);
END_TYPE;

TYPE si_prefix = ENUMERATION OF
     (exa,peta,tera,giga,mega,kilo,hecto,deca,deci,centi,milli,micro,nano,pico
      ,femto,atto);
END_TYPE;

TYPE si_unit_name = ENUMERATION OF
     (metre,gram,second,ampere,kelvin,mole,candela,radian,steradian,hertz,
      newton,pascal,joule,watt,coulomb,volt,farad,ohm,siemens,weber,tesla,
      henry,degree_celsius,lumen,lux,becquerel,gray,sievert);
END_TYPE;

TYPE size_select = SELECT
     (positive_length_measure,measure_with_unit,descriptive_measure);
END_TYPE;

TYPE solid_angle_measure = REAL;
END_TYPE;

TYPE source_item = SELECT
     (identifier);
END_TYPE;

TYPE spatial_rotation = SELECT
     (ypr_rotation,rotation_about_direction);
END_TYPE;

TYPE specified_item = SELECT
     (drawing_revision);
END_TYPE;
```

```
TYPE style_context_select = SELECT
     (representation,representation_item,presentation_set);
END_TYPE;


TYPE supported_item = SELECT
     (action_directive,action,action_method);
END_TYPE;


TYPE surface_model = SELECT
     (shell_based_surface_model,face_based_surface_model);
END_TYPE;


TYPE surface_side = ENUMERATION OF
     (positive,negative,both);
END_TYPE;


TYPE surface_side_style_select = SELECT
     (surface_side_style);
END_TYPE;


TYPE surface_style_element_select = SELECT
     (surface_style_fill_area,surface_style_boundary,surface_style_silhouette,
      surface_style_segmentation_curve,surface_style_control_grid,
      surface_style_parameter_line,surface_style_rendering);
END_TYPE;


TYPE symbol_style_select = SELECT
     (symbol_colour);
END_TYPE;


TYPE text = STRING;
END_TYPE;


TYPE text_alignment = label;
END_TYPE;


TYPE text_delineation = label;
END_TYPE;


TYPE text_or_character = SELECT
     (annotation_text,composite_text,text_literal);
END_TYPE;
```

```
TYPE text_path = ENUMERATION OF
     (left,right,up,down);
END_TYPE;


TYPE thermodynamic_temperature_measure = REAL;
END_TYPE;


TYPE time_measure = REAL;
END_TYPE;


TYPE tolerance_method_definition = SELECT
     (tolerance_value,limits_and_fits);
END_TYPE;


TYPE tolerance_select = SELECT
     (geometric_tolerance,plus_minus_tolerance);
END_TYPE;


TYPE transformation = SELECT
     (item_defined_transformation,functionally_defined_transformation);
END_TYPE;


TYPE transition_code = ENUMERATION OF
     (discontinuous,continuous,cont_same_gradient,
      cont_same_gradient_same_curvature);
END_TYPE;


TYPE translational_range_measure = SELECT
     (length_measure,unlimited_range);
END_TYPE;


TYPE trimming_preference = ENUMERATION OF
     (cartesian,parameter,unspecified);
END_TYPE;


TYPE trimming_select = SELECT
     (cartesian_point,parameter_value);
END_TYPE;


TYPE u_direction_count = INTEGER;
 WHERE
  wr1 : SELF > 1;
END_TYPE;
```

```
TYPE unit = SELECT
     (named_unit,derived_unit);
END_TYPE;


TYPE unlimited_range = ENUMERATION OF
     (unlimited);
END_TYPE;


TYPE v_direction_count = INTEGER;
 WHERE
  wr1 : SELF > 1;
END_TYPE;


TYPE value_qualifier = SELECT
     (precision_qualifier,type_qualifier,uncertainty_qualifier);
END_TYPE;


TYPE vector_or_direction = SELECT
     (vector,direction);
END_TYPE;


TYPE version_item = SELECT
     (action_directive,configurable_item,document,draughting_model,
      mechanical_design_geometric_presentation_representation,
      physically_modelled_product_definition,presentation_area,process_plan,
      product_class,product_component,product_concept_feature,
      product_definition,representation,shape_representation,
      solution_based_product_definition);
END_TYPE;


TYPE volume_measure = REAL;
END_TYPE;


TYPE wireframe_model = SELECT
     (edge_based_wireframe_model);
END_TYPE;


TYPE year_number = INTEGER;
END_TYPE;


TYPE ypr_enumeration = ENUMERATION OF
     (yaw,pitch,roll);
END_TYPE;
```

```
TYPE ypr_rotation = ARRAY[ypr_index(yaw):ypr_index(roll)] OF
      plane_angle_measure;
END_TYPE;


ENTITY action;
 id : identifier;
 name : label;
 description : text;
 chosen_method : action_method;
END_ENTITY;


ENTITY action_assignment
 ABSTRACT SUPERTYPE;
 assigned_action : action;
 role : action_role;
END_ENTITY;


ENTITY action_directive;
 name : label;
 description : text;
 analysis : text;
 comment : text;
 requests : SET[1:?] OF versioned_action_request;
END_ENTITY;


ENTITY action_method;
 name : label;
 description : text;
 consequence : text;
 purpose : text;
END_ENTITY;


ENTITY action_property;
 name : label;
 description : text;
 definition : characterized_action_definition;
END_ENTITY;


ENTITY action_property_representation;
 name : label;
 description : text;
 property : action_property;
 representation : representation;
END_ENTITY;


ENTITY action_relationship;
 name : label;
 description : text;
```

```
 relating_action : action;
 related_action : action;
END_ENTITY;

ENTITY action_request_assignment
 ABSTRACT SUPERTYPE;
 assigned_action_request : versioned_action_request;
 role : action_request_role;
END_ENTITY;

ENTITY action_request_role;
 name : label;
 description : text;
END_ENTITY;

ENTITY action_request_solution;
 name : label;
 description : text;
 method : action_method;
 request : versioned_action_request;
END_ENTITY;

ENTITY action_request_status;
 status : label;
 assigned_request : versioned_action_request;
END_ENTITY;

ENTITY action_resource;
 name : label;
 description : text;
 usage : SET[1:?] OF supported_item;
 kind : action_resource_type;
END_ENTITY;

ENTITY action_resource_requirement;
 name : label;
 description : text;
 kind : resource_requirement_type;
 operations : SET[1:?] OF characterized_action_definition;
END_ENTITY;

ENTITY action_resource_type;
 name : label;
END_ENTITY;

ENTITY action_role;
 name : label;
```

```
 description : text;
END_ENTITY;


ENTITY action_status;
 status : label;
 assigned_action : executed_action;
END_ENTITY;


ENTITY address;
 internal_location : OPTIONAL label;
 street_number : OPTIONAL label;
 street : OPTIONAL label;
 postal_box : OPTIONAL label;
 town : OPTIONAL label;
 region : OPTIONAL label;
 postal_code : OPTIONAL label;
 country : OPTIONAL label;
 facsimile_number : OPTIONAL label;
 telephone_number : OPTIONAL label;
 electronic_mail_address : OPTIONAL label;
 telex_number : OPTIONAL label;
 WHERE
  wr1 : EXISTS(internal_location) OR EXISTS(street_number) OR EXISTS(street)
        OR EXISTS(postal_box) OR EXISTS(town) OR EXISTS(region) OR EXISTS(
        postal_code) OR EXISTS(country) OR EXISTS(facsimile_number) OR EXISTS(
        telephone_number) OR EXISTS(electronic_mail_address) OR EXISTS(
        telex_number);
END_ENTITY;


ENTITY advanced_brep_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(it <* SELF.items | (NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.MANIFOLD_SOLID_BREP',
        'AUTOMOTIVE_DESIGN.FACETED_BREP','AUTOMOTIVE_DESIGN.MAPPED_ITEM',
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D'] * TYPEOF(it)) = 1)))) = 0;
  wr2 : SIZEOF(QUERY(it <* SELF.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MANIFOLD_SOLID_BREP',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM'] * TYPEOF(it)) = 1))) > 0;
  wr3 : SIZEOF(QUERY(msb <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.MANIFOLD_SOLID_BREP' IN TYPEOF(it))) | (NOT (SIZEOF
        (QUERY(csh <* msb_shells(msb) | (NOT (SIZEOF(QUERY(fcs <* csh\
        connected_face_set.cfs_faces | (NOT ('AUTOMOTIVE_DESIGN.ADVANCED_FACE'
         IN TYPEOF(fcs))))) = 0)))) = 0)))) = 0;
  wr4 : SIZEOF(QUERY(msb <* QUERY(it <* items | (
        'AUTOMOTIVE_DESIGN.MANIFOLD_SOLID_BREP' IN TYPEOF(it))) | (
        'AUTOMOTIVE_DESIGN.ORIENTED_CLOSED_SHELL' IN TYPEOF(msb\
        manifold_solid_brep.outer)))) = 0;
  wr5 : SIZEOF(QUERY(brv <* QUERY(it <* items | (
        'AUTOMOTIVE_DESIGN.BREP_WITH_VOIDS' IN TYPEOF(it))) | (NOT (SIZEOF(
```

```
        QUERY(csh <* brv\brep_with_voids.voids | (csh\oriented_closed_shell.
        orientation))) = 0)))) = 0;
  wr6 : SIZEOF(QUERY(mi <* QUERY(it <* items | (
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(it))) | (NOT (
        'AUTOMOTIVE_DESIGN.ADVANCED_BREP_SHAPE_REPRESENTATION' IN TYPEOF(mi\
        mapped_item.mapping_source.mapped_representation))))) = 0;
END_ENTITY;


ENTITY advanced_face
 SUBTYPE OF (face_surface);
 WHERE
  wr1 : SIZEOF(['AUTOMOTIVE_DESIGN.ELEMENTARY_SURFACE',
        'AUTOMOTIVE_DESIGN.B_SPLINE_SURFACE','AUTOMOTIVE_DESIGN.SWEPT_SURFACE'
        ] * TYPEOF(face_geometry)) = 1;
  wr2 : SIZEOF(QUERY(elp_fbnds <* QUERY(bnds <* bounds |
        'AUTOMOTIVE_DESIGN.EDGE_LOOP' IN TYPEOF(bnds.bound)) | NOT (SIZEOF(
        QUERY(oe <* elp_fbnds.bound\path.edge_list | NOT (
        'AUTOMOTIVE_DESIGN.EDGE_CURVE' IN TYPEOF(oe.edge_element)))) = 0))) =
        0;
  wr3 : SIZEOF(QUERY(elp_fbnds <* QUERY(bnds <* bounds |
        'AUTOMOTIVE_DESIGN.EDGE_LOOP' IN TYPEOF(bnds.bound)) | NOT (SIZEOF(
        QUERY(oe <* elp_fbnds.bound\path.edge_list | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.LINE','AUTOMOTIVE_DESIGN.CONIC',
        'AUTOMOTIVE_DESIGN.POLYLINE','AUTOMOTIVE_DESIGN.SURFACE_CURVE',
        'AUTOMOTIVE_DESIGN.B_SPLINE_CURVE'] * TYPEOF(oe.edge_element\
        edge_curve.edge_geometry)) = 1))) = 0))) = 0;
  wr4 : SIZEOF(QUERY(elp_fbnds <* QUERY(bnds <* bounds |
        'AUTOMOTIVE_DESIGN.EDGE_LOOP' IN TYPEOF(bnds.bound)) | NOT (SIZEOF(
        QUERY(oe <* elp_fbnds.bound\path.edge_list | NOT ((((
        'AUTOMOTIVE_DESIGN.VERTEX_POINT' IN TYPEOF(oe.edge_start)) AND (
        'AUTOMOTIVE_DESIGN.CARTESIAN_POINT' IN TYPEOF(oe.edge_start\
        vertex_point.vertex_geometry)))) AND ((
        'AUTOMOTIVE_DESIGN.VERTEX_POINT' IN TYPEOF(oe.edge_end)) AND (
        'AUTOMOTIVE_DESIGN.CARTESIAN_POINT' IN TYPEOF(oe.edge_end\vertex_point
        .vertex_geometry)))))) = 0))) = 0;
  wr5 : SIZEOF(QUERY(elp_fbnds <* QUERY(bnds <* bounds |
        'AUTOMOTIVE_DESIGN.EDGE_LOOP' IN TYPEOF(bnds.bound)) |
        'AUTOMOTIVE_DESIGN.ORIENTED_PATH' IN TYPEOF(elp_fbnds.bound))) = 0;
  wr6 : (NOT ('AUTOMOTIVE_DESIGN.SWEPT_SURFACE' IN TYPEOF(face_geometry))) OR
        (SIZEOF(['AUTOMOTIVE_DESIGN.LINE','AUTOMOTIVE_DESIGN.CONIC',
        'AUTOMOTIVE_DESIGN.POLYLINE','AUTOMOTIVE_DESIGN.B_SPLINE_CURVE'] *
        TYPEOF(face_geometry\swept_surface.swept_curve)) = 1);
  wr7 : SIZEOF(QUERY(vlp_fbnds <* QUERY(bnds <* bounds |
        'AUTOMOTIVE_DESIGN.VERTEX_LOOP' IN TYPEOF(bnds.bound)) | NOT ((
        'AUTOMOTIVE_DESIGN.VERTEX_POINT' IN TYPEOF(vlp_fbnds\face_bound.bound\
        vertex_loop.loop_vertex)) AND ('AUTOMOTIVE_DESIGN.CARTESIAN_POINT' IN
        TYPEOF(vlp_fbnds\face_bound.bound\vertex_loop.loop_vertex\vertex_point
        .vertex_geometry))))) = 0;
  wr8 : SIZEOF(QUERY(bnd <* bounds | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.EDGE_LOOP','AUTOMOTIVE_DESIGN.VERTEX_LOOP'] *
        TYPEOF(bnd.bound)) = 1))) = 0;
  wr9 : SIZEOF(QUERY(elp_fbnds <* QUERY(bnds <* bounds |
```

```
        'AUTOMOTIVE_DESIGN.EDGE_LOOP' IN TYPEOF(bnds.bound)) | NOT (SIZEOF(
        QUERY(oe <* elp_fbnds.bound\path.edge_list | (
        'AUTOMOTIVE_DESIGN.SURFACE_CURVE' IN TYPEOF(oe.edge_element\edge_curve
        .edge_geometry)) AND (NOT (SIZEOF(QUERY(sc_ag <* oe.edge_element\
        edge_curve.edge_geometry\surface_curve.associated_geometry | NOT (
        'AUTOMOTIVE_DESIGN.PCURVE' IN TYPEOF(sc_ag)))) = 0)))) = 0))) = 0;
  wr10 : ((NOT ('AUTOMOTIVE_DESIGN.SWEPT_SURFACE' IN TYPEOF(face_geometry)))
         OR ((NOT ('AUTOMOTIVE_DESIGN.POLYLINE' IN TYPEOF(face_geometry\
         swept_surface.swept_curve))) OR (SIZEOF(face_geometry\swept_surface.
         swept_curve\polyline.points) < 3))) AND (SIZEOF(QUERY(elp_fbnds <*
         QUERY(bnds <* bounds | 'AUTOMOTIVE_DESIGN.EDGE_LOOP' IN TYPEOF(bnds.
         bound)) | NOT (SIZEOF(QUERY(oe <* elp_fbnds.bound\path.edge_list | (
         'AUTOMOTIVE_DESIGN.POLYLINE' IN TYPEOF(oe.edge_element\edge_curve.
         edge_geometry)) AND (NOT (SIZEOF(oe.edge_element\edge_curve.
         edge_geometry\polyline.points) < 3)))) = 0))) = 0);
END_ENTITY;


ENTITY alternate_product_relationship;
 name : label;
 description : text;
 alternate : product;
 base : product;
 basis : text;
 UNIQUE
  ur1 : alternate,base;
 WHERE
  wr1 : alternate :<>: base;
END_ENTITY;


ENTITY amount_of_substance_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.AMOUNT_OF_SUBSTANCE_UNIT' IN TYPEOF(SELF\
        measure_with_unit.unit_component);
END_ENTITY;


ENTITY amount_of_substance_unit
 SUBTYPE OF (named_unit);
 WHERE
  wr1 : (SELF\named_unit.dimensions.length_exponent = 0.0) AND (SELF\
        named_unit.dimensions.mass_exponent = 0.0) AND (SELF\named_unit.
        dimensions.time_exponent = 0.0) AND (SELF\named_unit.dimensions.
        electric_current_exponent = 0.0) AND (SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 0.0) AND (SELF\named_unit.
        dimensions.amount_of_substance_exponent = 1.0) AND (SELF\named_unit.
        dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;


ENTITY angular_dimension
 SUBTYPE OF (dimension_curve_directed_callout);
```

```
END_ENTITY;


ENTITY angular_location
 SUBTYPE OF (dimensional_location);
 angle_selection : angle_relator;
END_ENTITY;


ENTITY angular_size
 SUBTYPE OF (dimensional_size);
 angle_selection : angle_relator;
END_ENTITY;


ENTITY angularity_tolerance
 SUBTYPE OF (geometric_tolerance_with_datum_reference);
 WHERE
  wr1 : SIZEOF(SELF\geometric_tolerance_with_datum_reference.datum_system) = 1
        ;
END_ENTITY;


ENTITY annotation_curve_occurrence
 SUBTYPE OF (annotation_occurrence);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF(SELF\styled_item.item);
END_ENTITY;


ENTITY annotation_fill_area
 SUBTYPE OF (geometric_representation_item);
 boundaries : SET[1:?] OF curve;
END_ENTITY;


ENTITY annotation_fill_area_occurrence
 SUBTYPE OF (annotation_occurrence);
 fill_style_target : point;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.ANNOTATION_FILL_AREA' IN TYPEOF(SELF.item);
END_ENTITY;


ENTITY annotation_occurrence
 SUPERTYPE OF (ONEOF(annotation_curve_occurrence,
               annotation_fill_area_occurrence,annotation_text_occurrence,
               annotation_symbol_occurrence))
 SUBTYPE OF (styled_item);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_ITEM' IN TYPEOF(SELF);
END_ENTITY;
```

```
ENTITY annotation_occurrence_associativity
 SUBTYPE OF (annotation_occurrence_relationship);
 WHERE
  wr1 : SIZEOF(TYPEOF(SELF.related_annotation_occurrence) * [
        'AUTOMOTIVE_DESIGN.ANNOTATION_FILL_AREA_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.PROJECTION_CURVE','AUTOMOTIVE_DESIGN.LEADER_CURVE']
        ) = 1;
END_ENTITY;


ENTITY annotation_occurrence_relationship;
 name : label;
 description : text;
 relating_annotation_occurrence : annotation_occurrence;
 related_annotation_occurrence : annotation_occurrence;
END_ENTITY;


ENTITY annotation_subfigure_occurrence
 SUBTYPE OF (annotation_symbol_occurrence);
 WHERE
  wr1 : SIZEOF(QUERY(sty <* SELF.styles | NOT (SIZEOF(sty.styles) = 1))) = 0;
  wr2 : SIZEOF(QUERY(sty <* SELF.styles | NOT ('AUTOMOTIVE_DESIGN.NULL_STYLE'
        IN TYPEOF(sty.styles[1])))) = 0;
  wr3 : ('AUTOMOTIVE_DESIGN.ANNOTATION_SYMBOL' IN TYPEOF(SELF.item));
  wr4 : ('AUTOMOTIVE_DESIGN.DRAUGHTING_SUBFIGURE_REPRESENTATION' IN TYPEOF(
        SELF.item\mapped_item.mapping_source.mapped_representation));
END_ENTITY;


ENTITY annotation_symbol
 SUBTYPE OF (mapped_item);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.SYMBOL_REPRESENTATION_MAP' IN TYPEOF(SELF\
        mapped_item.mapping_source);
  wr2 : 'AUTOMOTIVE_DESIGN.SYMBOL_TARGET' IN TYPEOF(SELF\mapped_item.
        mapping_target);
  wr3 : 'AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_ITEM' IN TYPEOF(SELF);
END_ENTITY;


ENTITY annotation_symbol_occurrence
 SUBTYPE OF (annotation_occurrence);
 WHERE
  wr1 : SIZEOF(['AUTOMOTIVE_DESIGN.ANNOTATION_SYMBOL',
        'AUTOMOTIVE_DESIGN.DEFINED_SYMBOL'] * TYPEOF(SELF\styled_item.item)) >
         0;
END_ENTITY;


ENTITY annotation_text
 SUBTYPE OF (mapped_item);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT' IN TYPEOF(SELF\mapped_item.
```

```
        mapping_target);
  wr2 : 'AUTOMOTIVE_DESIGN.TEXT_STRING_REPRESENTATION' IN TYPEOF(SELF\
        mapped_item.mapping_source.mapped_representation);
  wr3 : 'AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_ITEM' IN TYPEOF(SELF);
END_ENTITY;


ENTITY annotation_text_occurrence
 SUBTYPE OF (annotation_occurrence);
 WHERE
  wr1 : SIZEOF(['AUTOMOTIVE_DESIGN.TEXT_LITERAL',
        'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT',
        'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_CHARACTER',
        'AUTOMOTIVE_DESIGN.DEFINED_CHARACTER_GLYPH',
        'AUTOMOTIVE_DESIGN.COMPOSITE_TEXT'] * TYPEOF(SELF\styled_item.item)) >
         0;
END_ENTITY;


ENTITY apex
 SUBTYPE OF (derived_shape_aspect);
END_ENTITY;


ENTITY application_context;
 name : label;
 description : text;
 INVERSE
  context_elements : SET[1:?] OF application_context_element FOR
                     frame_of_reference;
END_ENTITY;


ENTITY application_context_element
 SUPERTYPE OF (ONEOF(product_context,product_definition_context,
               product_concept_context,configuration_item_context));
 name : label;
 frame_of_reference : SET[1:?] OF application_context;
END_ENTITY;


ENTITY application_context_element_relationship;
 name : label;
 description : text;
 relating_context_element : application_context_element;
 related_context_element : application_context_element;
END_ENTITY;


ENTITY application_protocol_definition;
 status : label;
 application_interpreted_model_schema_name : label;
 application_protocol_year : year_number;
 application : SET[1:?] OF application_context;
```

```
END_ENTITY;

ENTITY applied_action_assignment
 SUBTYPE OF (action_assignment);
 items : SET[1:?] OF action_item;
END_ENTITY;

ENTITY applied_action_request_assignment
 SUBTYPE OF (action_request_assignment);
 items : SET[1:?] OF action_request_item;
END_ENTITY;

ENTITY applied_approval_assignment
 SUBTYPE OF (approval_assignment);
 items : SET[1:?] OF approval_item;
END_ENTITY;

ENTITY applied_certification_assignment
 SUBTYPE OF (certification_assignment);
 items : SET[1:?] OF certification_item;
END_ENTITY;

ENTITY applied_contract_assignment
 SUBTYPE OF (contract_assignment);
 items : SET[1:?] OF contract_item;
END_ENTITY;

ENTITY applied_date_and_organization_assignment
 SUBTYPE OF (organization_assignment);
 items : SET[1:?] OF date_and_organization_item;
END_ENTITY;

ENTITY applied_date_and_person_and_organization_assignment
 SUBTYPE OF (person_and_organization_assignment);
 items : SET[1:?] OF date_and_person_and_organization_item;
END_ENTITY;

ENTITY applied_date_and_person_assignment
 SUBTYPE OF (person_assignment);
 items : SET[1:?] OF date_and_person_item;
END_ENTITY;

ENTITY applied_date_and_time_assignment
 SUBTYPE OF (date_and_time_assignment);
 items : SET[1:?] OF date_and_time_item;
END_ENTITY;
```

```
ENTITY applied_date_assignment
 SUBTYPE OF (date_assignment);
 items : SET[1:?] OF date_item;
END_ENTITY;


ENTITY applied_document_reference
 SUBTYPE OF (document_reference);
 items : SET[1:?] OF document_reference_item;
END_ENTITY;


ENTITY applied_effective_context_assignment
 SUBTYPE OF (effective_context_assignment);
 items : SET[1:?] OF effective_context_item;
 WHERE
  wr1 : SIZEOF(['AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY',
        'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY'] * TYPEOF(SELF.
        assigned_effectivity)) = 1;
  wr2 : (SIZEOF(QUERY(i <* SELF.items | 'AUTOMOTIVE_DESIGN.ORGANIZATION' IN
        TYPEOF(i))) = 0) OR (SELF.role.name = 'concerned organization');
  wr3 : (SIZEOF(QUERY(i <* SELF.items |
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_CONTEXT' IN TYPEOF(i))) = 0) OR
        (SELF.role.name = 'effectivity context');
END_ENTITY;


ENTITY applied_effective_item_assignment
 SUBTYPE OF (effective_item_assignment);
 items : SET[1:?] OF effective_item;
 WHERE
  wr1 : SIZEOF(['AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY',
        'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY'] * TYPEOF(SELF.
        assigned_effectivity)) = 1;
END_ENTITY;


ENTITY applied_external_identification_assignment
 SUBTYPE OF (external_identification_assignment);
 items : SET[1:?] OF external_identification_item;
END_ENTITY;


ENTITY applied_group_assignment
 SUBTYPE OF (group_assignment);
 items : SET[1:?] OF group_item;
END_ENTITY;


ENTITY applied_identification_assignment
 SUBTYPE OF (identification_assignment);
 items : SET[1:?] OF identification_item;
END_ENTITY;
```

```
ENTITY applied_multi_language_attribute_assignment
 SUBTYPE OF (multi_language_attribute_assignment);
 items : SET[1:?] OF multi_language_attribute_item;
END_ENTITY;


ENTITY applied_organization_assignment
 SUBTYPE OF (organization_assignment);
 items : SET[1:?] OF organization_item;
END_ENTITY;


ENTITY applied_person_and_organization_assignment
 SUBTYPE OF (person_and_organization_assignment);
 items : SET[1:?] OF person_and_organization_item;
END_ENTITY;


ENTITY applied_person_assignment
 SUBTYPE OF (person_assignment);
 items : SET[1:?] OF person_item;
END_ENTITY;


ENTITY applied_presented_item
 SUBTYPE OF (presented_item);
 items : SET[1:?] OF presented_item_select;
END_ENTITY;


ENTITY applied_security_classification_assignment
 SUBTYPE OF (security_classification_assignment);
 items : SET[1:?] OF security_classification_item;
END_ENTITY;


ENTITY approval;
 status : approval_status;
 level : label;
END_ENTITY;


ENTITY approval_assignment
 ABSTRACT SUPERTYPE;
 assigned_approval : approval;
 role : approval_role;
END_ENTITY;


ENTITY approval_date_time;
 date_time : date_time_select;
 dated_approval : approval;
 role : approval_date_time_role;
END_ENTITY;
```

```
ENTITY approval_date_time_role;
 name : label;
 description : text;
END_ENTITY;


ENTITY approval_person_organization;
 person_organization : person_organization_select;
 authorized_approval : approval;
 role : approver_role;
END_ENTITY;


ENTITY approval_relationship;
 name : label;
 description : text;
 relating_approval : approval;
 related_approval : approval;
END_ENTITY;


ENTITY approval_role;
 name : label;
 description : text;
END_ENTITY;


ENTITY approval_status;
 name : label;
END_ENTITY;


ENTITY approver_role;
 name : label;
 description : text;
END_ENTITY;


ENTITY area_in_set;
 area : presentation_area;
 in_set : presentation_set;
END_ENTITY;


ENTITY area_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.AREA_UNIT' IN TYPEOF(SELF\measure_with_unit.
        unit_component);
END_ENTITY;


ENTITY area_presentation_relationship
 SUBTYPE OF (representation_relationship);
 WHERE
```

```
  wr1 : ('AUTOMOTIVE_DESIGN.PRESENTATION_AREA' IN TYPEOF(SELF\
        representation_relationship.rep_1)) AND (
        'AUTOMOTIVE_DESIGN.PRESENTATION_AREA' IN TYPEOF(SELF\
        representation_relationship.rep_2));
END_ENTITY;


ENTITY area_unit
 SUBTYPE OF (named_unit);
 WHERE
  wr1 : (SELF\named_unit.dimensions.length_exponent = 2.0) AND (SELF\
        named_unit.dimensions.mass_exponent = 0.0) AND (SELF\named_unit.
        dimensions.time_exponent = 0.0) AND (SELF\named_unit.dimensions.
        electric_current_exponent = 0.0) AND (SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 0.0) AND (SELF\named_unit.
        dimensions.amount_of_substance_exponent = 0.0) AND (SELF\named_unit.
        dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;


ENTITY assembly_component_usage
 SUPERTYPE OF (ONEOF(next_assembly_usage_occurrence,
               specified_higher_usage_occurrence,promissory_usage_occurrence))
 SUBTYPE OF (product_definition_usage);
 reference_designator : OPTIONAL identifier;
END_ENTITY;


ENTITY assembly_component_usage_substitute;
 name : label;
 description : text;
 ranking : INTEGER;
 ranking_rationale : text;
 base : assembly_component_usage;
 substitute : assembly_component_usage;
 UNIQUE
  ur1 : base,substitute;
 WHERE
  wr1 : base.relating_product_definition :=: substitute.
        relating_product_definition;
  wr2 : base :<>: substitute;
END_ENTITY;


ENTITY auto_design_drawing_revision
 SUBTYPE OF (drawing_revision);
 WHERE
  wr1 : (SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET')) >= 1) AND
         (SIZEOF(QUERY(ais <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') | NOT (
        'AUTOMOTIVE_DESIGN.DRAWING_SHEET_REVISION' IN TYPEOF(ais.area)))) = 0)
        ;
  wr2 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'DRAUGHTING_TITLE.ITEMS'))
        <= 1;
```

```
wr3 : SIZEOF(QUERY(ais <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET'
      ) | NOT (SIZEOF(USEDIN(ais.area,'AUTOMOTIVE_DESIGN.' +
      'DRAUGHTING_TITLE.ITEMS')) <= 1))) = 0;
wr4 : SIZEOF(QUERY(ais <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET'
      ) | NOT (SIZEOF(QUERY(item <* ais.area.items | NOT (SIZEOF(TYPEOF(item
      ) * ['AUTOMOTIVE_DESIGN.STYLED_ITEM','AUTOMOTIVE_DESIGN.MAPPED_ITEM',
      'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT','AUTOMOTIVE_DESIGN.PLANAR_BOX']) =
       1))) = 0))) = 0;
wr5 : SIZEOF(QUERY(ais <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET'
      ) | NOT (SIZEOF(QUERY(item <* ais.area.items | (SIZEOF(TYPEOF(item) *
      ['AUTOMOTIVE_DESIGN.STYLED_ITEM','AUTOMOTIVE_DESIGN.MAPPED_ITEM']) = 1
      ))) > 0))) = 0;
wr6 : SIZEOF(QUERY(ais <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET'
      ) | NOT (SIZEOF(QUERY(p_b <* QUERY(item <* ais.area.items | (
      'AUTOMOTIVE_DESIGN.PLANAR_BOX' IN TYPEOF(item))) | NOT (SIZEOF(USEDIN(
      p_b,'AUTOMOTIVE_DESIGN.PRESENTATION_SIZE.SIZE')) = 1))) = 0))) = 0;
wr7 : SIZEOF(QUERY(ais <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET'
      ) | NOT (SIZEOF(QUERY(mi <* QUERY(item <* ais.area.items | (
      'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(item))) | NOT (
      'AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN TYPEOF(mi.mapping_source.
      mapped_representation)))) = 0))) = 0;
wr8 : SIZEOF(QUERY(ais <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET'
      ) | NOT (SIZEOF(QUERY(a2p <* QUERY(item <* ais.area.items | (
      'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT' IN TYPEOF(item))) | NOT (SIZEOF(
      USEDIN(a2p,'AUTOMOTIVE_DESIGN.' + 'MAPPED_ITEM.MAPPING_TARGET')) > 0))
      ) = 0))) = 0;
wr9 : SIZEOF(QUERY(ais <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET'
      ) | NOT (SIZEOF(ais.area.context_of_items.representations_in_context)
      = 1))) = 0;
wr10 : SIZEOF(QUERY(ais <* USEDIN(SELF,
       'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') | NOT (SIZEOF(QUERY(mi <*
       QUERY(item <* ais.area.items | ('AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN
       TYPEOF(item))) | NOT (SIZEOF(USEDIN(mi.mapping_source.
       mapped_representation,'AUTOMOTIVE_DESIGN.' +
       'REPRESENTATION_MAP.MAPPED_REPRESENTATION')) = 1))) = 0))) = 0;
wr11 : SIZEOF(QUERY(ais <* USEDIN(SELF,
       'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') | NOT (SIZEOF(QUERY(mi <*
       QUERY(item <* ais.area.items | ('AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN
       TYPEOF(item))) | NOT (SIZEOF(QUERY(pv_item <* mi.mapping_source.
       mapped_representation.items | NOT (SIZEOF(TYPEOF(pv_item) * [
       'AUTOMOTIVE_DESIGN.' + 'STYLED_ITEM','AUTOMOTIVE_DESIGN.CAMERA_IMAGE'
       ,'AUTOMOTIVE_DESIGN.' + 'AXIS2_PLACEMENT']) = 1))) = 0))) = 0))) = 0;
wr12 : SIZEOF(QUERY(ais <* USEDIN(SELF,
       'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') | NOT (SIZEOF(QUERY(mi <*
       QUERY(item <* ais.area.items | ('AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN
       TYPEOF(item))) | NOT (SIZEOF(QUERY(pv_item <* mi.mapping_source.
       mapped_representation.items | ('AUTOMOTIVE_DESIGN.CAMERA_IMAGE' IN
       TYPEOF(pv_item)))) = 1))) = 0))) = 0;
wr13 : SIZEOF(QUERY(ais <* USEDIN(SELF,
       'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') | NOT (SIZEOF(QUERY(mi <*
       QUERY(item <* ais.area.items | ('AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN
       TYPEOF(item))) | NOT (SIZEOF(QUERY(a2p <* QUERY(pv_item <* mi.
```

```
        mapping_source.mapped_representation.items | (
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT' IN TYPEOF(pv_item))) | (
        'AUTOMOTIVE_DESIGN.' + 'MAPPING_SOURCE.MAPPING_ORIGIN' IN ROLESOF(a2p
        )))) = 1))) = 0))) = 0;
  wr14 : SIZEOF(QUERY(ais <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.AREA_IN_SET.IN_SET') | NOT (SIZEOF(QUERY(mi <*
        QUERY(item <* ais.area.items | ('AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN
        TYPEOF(item))) | NOT (SIZEOF(mi.mapping_source.mapped_representation.
        context_of_items.representations_in_context) = 1))) = 0))) = 0;
END_ENTITY;


ENTITY axis1_placement
 SUBTYPE OF (placement);
 axis : OPTIONAL direction;
 DERIVE
  z : direction := NVL(normalise(axis),direction([0.0,0.0,1.0]));
 WHERE
  wr1 : SELF\geometric_representation_item.dim = 3;
END_ENTITY;


ENTITY axis2_placement_2d
 SUBTYPE OF (placement);
 ref_direction : OPTIONAL direction;
 DERIVE
  p : LIST[2:2] OF direction := build_2axes(ref_direction);
 WHERE
  wr1 : SELF\geometric_representation_item.dim = 2;
END_ENTITY;


ENTITY axis2_placement_3d
 SUBTYPE OF (placement);
 axis : OPTIONAL direction;
 ref_direction : OPTIONAL direction;
 DERIVE
  p : LIST[3:3] OF direction := build_axes(axis,ref_direction);
 WHERE
  wr1 : SELF\placement.location.dim = 3;
  wr2 : (NOT (EXISTS(axis))) OR (axis.dim = 3);
  wr3 : (NOT (EXISTS(ref_direction))) OR (ref_direction.dim = 3);
  wr4 : (NOT (EXISTS(axis))) OR (NOT (EXISTS(ref_direction))) OR (
        cross_product(axis,ref_direction).magnitude > 0.0);
END_ENTITY;


ENTITY b_spline_curve
 SUPERTYPE OF (ONEOF(uniform_curve,b_spline_curve_with_knots,
               quasi_uniform_curve,bezier_curve) ANDOR rational_b_spline_curve
               )
 SUBTYPE OF (bounded_curve);
 degree : INTEGER;
 control_points_list : LIST[2:?] OF cartesian_point;
```

```
 curve_form : b_spline_curve_form;
 closed_curve : LOGICAL;
 self_intersect : LOGICAL;
 DERIVE
  upper_index_on_control_points : INTEGER := (SIZEOF(control_points_list) - 1)
                                  ;
  control_points : ARRAY[0:upper_index_on_control_points] OF cartesian_point :=
                   list_to_array(control_points_list,0,
                   upper_index_on_control_points);
 WHERE
  wr1 : ('AUTOMOTIVE_DESIGN.UNIFORM_CURVE' IN TYPEOF(SELF)) OR (
        'AUTOMOTIVE_DESIGN.QUASI_UNIFORM_CURVE' IN TYPEOF(SELF)) OR (
        'AUTOMOTIVE_DESIGN.BEZIER_CURVE' IN TYPEOF(SELF)) OR (
        'AUTOMOTIVE_DESIGN.B_SPLINE_CURVE_WITH_KNOTS' IN TYPEOF(SELF));
END_ENTITY;


ENTITY b_spline_curve_with_knots
 SUBTYPE OF (b_spline_curve);
 knot_multiplicities : LIST[2:?] OF INTEGER;
 knots : LIST[2:?] OF parameter_value;
 knot_spec : knot_type;
 DERIVE
  upper_index_on_knots : INTEGER := SIZEOF(knots);
 WHERE
  wr1 : constraints_param_b_spline(degree,upper_index_on_knots,
        upper_index_on_control_points,knot_multiplicities,knots);
  wr2 : SIZEOF(knot_multiplicities) = upper_index_on_knots;
END_ENTITY;


ENTITY b_spline_surface
 SUPERTYPE OF (ONEOF(b_spline_surface_with_knots,uniform_surface,
               quasi_uniform_surface,bezier_surface) ANDOR
               rational_b_spline_surface)
 SUBTYPE OF (bounded_surface);
 u_degree : INTEGER;
 v_degree : INTEGER;
 control_points_list : LIST[2:?] OF LIST[2:?] OF cartesian_point;
 surface_form : b_spline_surface_form;
 u_closed : LOGICAL;
 v_closed : LOGICAL;
 self_intersect : LOGICAL;
 DERIVE
  u_upper : INTEGER := SIZEOF(control_points_list) - 1;
  v_upper : INTEGER := SIZEOF(control_points_list[1]) - 1;
  control_points : ARRAY[0:u_upper] OF ARRAY[0:v_upper] OF cartesian_point :=
                   make_array_of_array(control_points_list,0,u_upper,0,v_upper
                   );
 WHERE
  wr1 : ('AUTOMOTIVE_DESIGN.UNIFORM_SURFACE' IN TYPEOF(SELF)) OR (
        'AUTOMOTIVE_DESIGN.QUASI_UNIFORM_SURFACE' IN TYPEOF(SELF)) OR (
        'AUTOMOTIVE_DESIGN.BEZIER_SURFACE' IN TYPEOF(SELF)) OR (
```

```
        'AUTOMOTIVE_DESIGN.B_SPLINE_SURFACE_WITH_KNOTS' IN TYPEOF(SELF));
END_ENTITY;


ENTITY b_spline_surface_with_knots
 SUBTYPE OF (b_spline_surface);
 u_multiplicities : LIST[2:?] OF INTEGER;
 v_multiplicities : LIST[2:?] OF INTEGER;
 u_knots : LIST[2:?] OF parameter_value;
 v_knots : LIST[2:?] OF parameter_value;
 knot_spec : knot_type;
 DERIVE
  knot_u_upper : INTEGER := SIZEOF(u_knots);
  knot_v_upper : INTEGER := SIZEOF(v_knots);
 WHERE
  wr1 : constraints_param_b_spline(SELF\b_spline_surface.u_degree,knot_u_upper
        ,SELF\b_spline_surface.u_upper,u_multiplicities,u_knots);
  wr2 : constraints_param_b_spline(SELF\b_spline_surface.v_degree,knot_v_upper
        ,SELF\b_spline_surface.v_upper,v_multiplicities,v_knots);
  wr3 : SIZEOF(u_multiplicities) = knot_u_upper;
  wr4 : SIZEOF(v_multiplicities) = knot_v_upper;
END_ENTITY;


ENTITY background_colour
 SUBTYPE OF (colour);
 presentation : area_or_view;
 UNIQUE
  ur1 : presentation;
END_ENTITY;


ENTITY bead
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND ({1<=SIZEOF(pdr.
        used_representation)<=3}))) = 1)) = 1;
  wr2 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | srwp_i.name IN ['orientation',
        'maximum depth','base radius'])) = SIZEOF(pdr.used_representation.
        items)))) = 1)) = 1;
  wr3 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
```

```
      TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (SIZEOF([
      'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
      AND (srwp_i.name = 'maximum depth'))) <= 1))) = 1)) = 1;
wr4 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (SIZEOF([
      'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
      AND (srwp_i.name = 'base radius'))) <= 1))) = 1)) = 1;
wr5 : SIZEOF(QUERY(sa <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'sweep path') AND (sa.
      description = 'path feature component occurrence') AND (SIZEOF(QUERY(
      sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar))) = 1)
      AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
      'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar)) AND (
      sar.name = 'path feature component usage') AND (
      'AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN TYPEOF(sar.
      relating_shape_aspect)) AND (sar.relating_shape_aspect.description =
      'linear'))) = 1))) = 1;
wr6 : SIZEOF(QUERY(sa <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'bead profile') AND (sa.
      description = 'open profile occurrence') AND (SIZEOF(QUERY(sar <*
      USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar))) = 1)
      AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
      'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar)) AND (
      sar.name = 'open profile usage') AND (SIZEOF([
      'AUTOMOTIVE_DESIGN.OPEN_PATH_PROFILE',
      'AUTOMOTIVE_DESIGN.PARTIAL_CIRCULAR_PROFILE',
      'AUTOMOTIVE_DESIGN.ROUNDED_U_PROFILE',
      'AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE','AUTOMOTIVE_DESIGN.TEE_PROFILE',
      'AUTOMOTIVE_DESIGN.VEE_PROFILE'] * TYPEOF(sar.relating_shape_aspect))
      = 1))) = 1))) = 1;
wr7 : SIZEOF(QUERY(sa <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'end condition at path start')
      AND (sa.description = 'bead end occurrence') AND (SIZEOF(QUERY(sar <*
      USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar))) =
      1) AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
      'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)) AND
```

```
         (sar.name = 'bead end usage') AND ('AUTOMOTIVE_DESIGN.BEAD_END' IN
        TYPEOF(sar.relating_shape_aspect)))) = 1))) = 1;
  wr8 : SIZEOF(QUERY(sa <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'end condition at path end') AND
         (sa.description = 'bead end occurrence') AND (SIZEOF(QUERY(sar <*
        USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar))) =
        1) AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)) AND
         (sar.name = 'bead end usage') AND ('AUTOMOTIVE_DESIGN.BEAD_END' IN
        TYPEOF(sar.relating_shape_aspect)))) = 1))) = 1;
END_ENTITY;


ENTITY bead_end
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SELF.description IN ['linear','open','radiused'];
  wr3 : (SELF.description <> 'linear') XOR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | 'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF
        (pdr.used_representation))) > 0);
  wr4 : (SELF.description <> 'linear') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.used_representation.
        items) = 2))) = 1);
  wr5 : (SELF.description <> 'linear') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'end length') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1);
  wr6 : (SELF.description <> 'linear') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'end shape') AND (srwp_i.
        description IN ['straight','tapered']) AND (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i))
        )) = 1))) = 1);
  wr7 : SIZEOF(QUERY(fcr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr)) AND
         (fcr.name = 'bead end usage') AND ('AUTOMOTIVE_DESIGN.BEAD' IN TYPEOF
        (fcr.related_shape_aspect.of_shape.definition)) AND (fcr.
        related_shape_aspect.of_shape.definition.name IN [
```

```
        'end condition at path start','end condition at path end']) AND (fcr.
        related_shape_aspect.of_shape.definition.description =
        'bead end occurrence'))) = 1;
END_ENTITY;


ENTITY bezier_curve
 SUBTYPE OF (b_spline_curve);
END_ENTITY;


ENTITY bezier_surface
 SUBTYPE OF (b_spline_surface);
END_ENTITY;


ENTITY block
 SUBTYPE OF (geometric_representation_item);
 position : axis2_placement_3d;
 x : positive_length_measure;
 y : positive_length_measure;
 z : positive_length_measure;
END_ENTITY;


ENTITY boolean_result
 SUBTYPE OF (geometric_representation_item);
 operator : boolean_operator;
 first_operand : boolean_operand;
 second_operand : boolean_operand;
END_ENTITY;


ENTITY boss
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SELF.description IN ['circular','rectangular','general'];
  wr2 : (SELF.description <> 'circular') OR (SIZEOF(QUERY(pds <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'boundary') AND (sa.description
        = 'closed profile occurrence') AND (SIZEOF(QUERY(sdr <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' + 'SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sdr)) AND (
        sdr.name = 'closed profile occurrence') AND (
        'AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE' IN TYPEOF(sdr.
        relating_shape_aspect)))) = 1))) = 1))) = 1);
  wr3 : (SELF.description <> 'general') OR (SIZEOF(QUERY(pds <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'boundary') AND (sa.description
```

```
      = 'closed profile occurrence') AND (SIZEOF(QUERY(sdr <* USEDIN(sa,
      'AUTOMOTIVE_DESIGN.' + 'SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | (
      'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sdr)) AND
       (sdr.name = 'closed profile usage') AND (
      'AUTOMOTIVE_DESIGN.CLOSED_PATH_PROFILE' IN TYPEOF(sdr.
      relating_shape_aspect)))) = 1))) = 1))) = 1);
wr4 : (SELF.description <> 'rectangular') OR (SIZEOF(QUERY(pds <* USEDIN(
      SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
      SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'boundary') AND (sa.description
      = 'closed profile occurrence') AND (SIZEOF(QUERY(sdr <* USEDIN(sa,
      'AUTOMOTIVE_DESIGN.' + 'SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | (
      'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sdr)) AND (
      sdr.description = 'closed profile usage') AND (
      'AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE' IN TYPEOF(sdr.
      relating_shape_aspect)))) = 1))) = 1))) = 1);
wr5 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.' +
      'DEFINITION') | (
      'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
      .used_representation)) AND ({1<=SIZEOF(pdr.used_representation.items)
      <=2}))) = 1)) = 1;
wr6 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.' +
      'DEFINITION') | (
      'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
      .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (srwp_i.name = 'orientation') OR (srwp_i.
      name = 'fillet radius'))) = SIZEOF(pdr.used_representation.items)))) =
       1)) = 1;
wr7 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.' +
      'DEFINITION') | (
      'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
      .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (srwp_i.name = 'fillet radius'))) <= 1)
      AND ((SIZEOF(QUERY(srwp_i <* pdr.used_representation.items | (srwp_i.
      name = 'fillet radius'))) = 0) OR (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (srwp_i.name = 'fillet radius') AND (
      SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
       = 1)))) = 1)) = 1;
wr8 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
      SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
```

```
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'sweep path') AND (sa.
        description = 'path feature component occurrence') AND (SIZEOF(QUERY(
        sdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' + 'SHAPE_ASPECT_RELATIONSHIP.' +
         'RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sdr)) AND (
        sdr.name = 'path feature component occurrence') AND (
        'AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN TYPEOF(sdr.
        relating_shape_aspect)) AND (sdr.relating_shape_aspect.description =
        'linear'))) = 1))) = 1))) = 1;
  wr9 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'top condition') AND (sa.
        description = 'boss top occurrence') AND (SIZEOF(QUERY(sdr <* USEDIN(
        sa,'AUTOMOTIVE_DESIGN.' + 'SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sdr)) AND
         (sdr.name = 'boss top usage') AND ('AUTOMOTIVE_DESIGN.BOSS_TOP' IN
        TYPEOF(sdr.relating_shape_aspect)))) = 1))) = 1))) = 1;
  wr10 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION.DEFINITION') | (
         'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
         SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
         'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'taper') AND (sa.description =
         'taper occurrence') AND (SIZEOF(QUERY(sdr <* USEDIN(sa,
         'AUTOMOTIVE_DESIGN.' + 'SHAPE_ASPECT_RELATIONSHIP.' +
         'RELATED_SHAPE_ASPECT') | (
         'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sdr))
         AND (sdr.name = 'taper usage') AND ('AUTOMOTIVE_DESIGN.TAPER' IN
         TYPEOF(sdr.relating_shape_aspect)))) = 1))) <= 1))) <= 1;
END_ENTITY;


ENTITY boss_top
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SELF.description IN ['planar','non-planar'];
  wr3 : (SELF.description <> 'planar') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | 'AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation))) = 1);
  wr4 : (SELF.description <> 'planar') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | 'AUTOMOTIVE_DESIGN.LOCATION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation))) = 1);
  wr5 : (SELF.description <> 'non-planar') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF
        ,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation)) OR (
```

```
        'AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation)))) = 1);
  wr6 : SIZEOF(QUERY(fcr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr)) AND
         (fcr.name = 'boss top usage') AND ('AUTOMOTIVE_DESIGN.BOSS' IN TYPEOF
        (fcr.related_shape_aspect.of_shape.definition)) AND (fcr.
        related_shape_aspect.name = 'top condition') AND (fcr.
        related_shape_aspect.description = 'boss top occurrence'))) = 0;
  wr7 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1;
  wr8 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(pdr.used_representation.items) = 1)
        )) = 1;
  wr9 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i))
         AND (srwp_i.name = 'top location') AND (srwp_i.description IN [
        'part start','part end']))) = 1))) = 1;
END_ENTITY;


ENTITY boundary_curve
 SUBTYPE OF (composite_curve_on_surface);
 WHERE
  wr1 : SELF\composite_curve.closed_curve;
END_ENTITY;


ENTITY bounded_curve
 SUPERTYPE OF (ONEOF(polyline,b_spline_curve,trimmed_curve,composite_curve))
 SUBTYPE OF (curve);
END_ENTITY;


ENTITY bounded_surface
 SUPERTYPE OF (ONEOF(b_spline_surface,rectangular_trimmed_surface,
               curve_bounded_surface,rectangular_composite_surface))
 SUBTYPE OF (surface);
END_ENTITY;


ENTITY box_domain;
 corner : cartesian_point;
 xlength : positive_length_measure;
 ylength : positive_length_measure;
 zlength : positive_length_measure;
```

```
 WHERE
  wr1 : SIZEOF(QUERY(item <* USEDIN(SELF,'') | NOT (
        'AUTOMOTIVE_DESIGN.BOXED_HALF_SPACE' IN TYPEOF(item)))) = 0;
END_ENTITY;


ENTITY boxed_half_space
 SUBTYPE OF (half_space_solid);
 enclosure : box_domain;
END_ENTITY;


ENTITY brep_with_voids
 SUBTYPE OF (manifold_solid_brep);
 voids : SET[1:?] OF oriented_closed_shell;
END_ENTITY;


ENTITY calendar_date
 SUBTYPE OF (date);
 day_component : day_in_month_number;
 month_component : month_in_year_number;
 WHERE
  wr1 : valid_calendar_date(SELF);
END_ENTITY;


ENTITY camera_image
 SUBTYPE OF (mapped_item);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.CAMERA_USAGE' IN TYPEOF(SELF\mapped_item.
        mapping_source);
  wr2 : 'AUTOMOTIVE_DESIGN.PLANAR_BOX' IN TYPEOF(SELF\mapped_item.
        mapping_target);
  wr3 : 'AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_ITEM' IN TYPEOF(SELF);
END_ENTITY;


ENTITY camera_image_2d_with_scale
 SUBTYPE OF (camera_image);
 DERIVE
  scale : positive_ratio_measure := ((SELF\mapped_item.mapping_target\
          planar_extent.size_in_x) / (SELF\mapped_item.mapping_source.
          mapping_origin\camera_model_d2.view_window.size_in_x));
 WHERE
  wr1 : ('AUTOMOTIVE_DESIGN.CAMERA_MODEL_D2' IN TYPEOF(SELF\mapped_item.
        mapping_source.mapping_origin));
  wr2 : aspect_ratio(SELF\mapped_item.mapping_target) = aspect_ratio(SELF\
        mapped_item.mapping_source.mapping_origin\camera_model_d2.view_window)
        ;
  wr3 : SELF\mapped_item.mapping_source.mapping_origin\camera_model_d2.
        view_window_clipping;
END_ENTITY;
```

```
ENTITY camera_image_3d_with_scale
 SUBTYPE OF (camera_image);
 DERIVE
  scale : positive_ratio_measure := ((SELF\mapped_item.mapping_target\
          planar_extent.size_in_x) / (SELF\mapped_item.mapping_source.
          mapping_origin\camera_model_d3.perspective_of_volume.view_window.
          size_in_x));
 WHERE
  wr1 : ('AUTOMOTIVE_DESIGN.CAMERA_MODEL_D3' IN TYPEOF(SELF\mapped_item.
        mapping_source.mapping_origin));
  wr2 : aspect_ratio(SELF\mapped_item.mapping_target) = aspect_ratio(SELF\
        mapped_item.mapping_source.mapping_origin\camera_model_d3.
        perspective_of_volume.view_window);
  wr3 : SELF\mapped_item.mapping_source.mapping_origin\camera_model_d3.
        perspective_of_volume.front_plane_clipping AND SELF\mapped_item.
        mapping_source.mapping_origin\camera_model_d3.perspective_of_volume.
        view_volume_sides_clipping;
END_ENTITY;


ENTITY camera_model
 SUPERTYPE OF (ONEOF(camera_model_d2,camera_model_d3))
 SUBTYPE OF (geometric_representation_item);
 WHERE
  wr1 : (SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ITEM_DEFINED_TRANSFORMATION.' + 'TRANSFORM_ITEM_1')) + SIZEOF(USEDIN(
        SELF,'AUTOMOTIVE_DESIGN.' + 'REPRESENTATION_MAP.MAPPING_ORIGIN'))) > 0
        ;
  wr2 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'STYLED_ITEM.ITEM')) = 0;
END_ENTITY;


ENTITY camera_model_d2
 SUBTYPE OF (camera_model);
 view_window : planar_box;
 view_window_clipping : BOOLEAN;
 WHERE
  wr1 : SELF\geometric_representation_item.dim = 2;
END_ENTITY;


ENTITY camera_model_d3
 SUBTYPE OF (camera_model);
 view_reference_system : axis2_placement_3d;
 perspective_of_volume : view_volume;
 WHERE
  wr1 : (dot_product(SELF.view_reference_system.p[3],SELF.
        perspective_of_volume.view_window.placement.p[3]) = 1.0) AND (SELF.
        view_reference_system.location.coordinates[3] = SELF.
        perspective_of_volume.view_window.placement.location.coordinates[3]);
  wr2 : SELF\geometric_representation_item.dim = 3;
END_ENTITY;
```

```
ENTITY camera_model_d3_with_hlhsr
 SUBTYPE OF (camera_model_d3);
 hidden_line_surface_removal : BOOLEAN;
END_ENTITY;


ENTITY camera_usage
 SUBTYPE OF (representation_map);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.PRESENTATION_REPRESENTATION' IN TYPEOF(SELF\
        representation_map.mapped_representation));
  wr2 : 'AUTOMOTIVE_DESIGN.CAMERA_MODEL' IN TYPEOF(SELF\representation_map.
        mapping_origin);
END_ENTITY;


ENTITY cartesian_point
 SUBTYPE OF (point);
 coordinates : LIST[1:3] OF length_measure;
END_ENTITY;


ENTITY cartesian_transformation_operator
 SUPERTYPE OF (ONEOF(cartesian_transformation_operator_2d,
               cartesian_transformation_operator_3d))
 SUBTYPE OF (geometric_representation_item,functionally_defined_transformation
             );
 axis1 : OPTIONAL direction;
 axis2 : OPTIONAL direction;
 local_origin : cartesian_point;
 scale : OPTIONAL REAL;
 DERIVE
  scl : REAL := NVL(scale,1.0);
 WHERE
  wr1 : scl > 0.0;
END_ENTITY;


ENTITY cartesian_transformation_operator_2d
 SUBTYPE OF (cartesian_transformation_operator);
 DERIVE
  u : LIST[2:2] OF direction := base_axis(2,SELF\
      cartesian_transformation_operator.axis1,SELF\
      cartesian_transformation_operator.axis2,?);
 WHERE
  wr1 : SELF\cartesian_transformation_operator.dim = 2;
END_ENTITY;


ENTITY cartesian_transformation_operator_3d
 SUBTYPE OF (cartesian_transformation_operator);
 axis3 : OPTIONAL direction;
 DERIVE
  u : LIST[3:3] OF direction := base_axis(3,SELF\
```

```
      cartesian_transformation_operator.axis1,SELF\
      cartesian_transformation_operator.axis2,axis3);
 WHERE
  wr1 : SELF\cartesian_transformation_operator.dim = 3;
END_ENTITY;


ENTITY centre_of_symmetry
 SUBTYPE OF (derived_shape_aspect);
 WHERE
  wr1 : SIZEOF(QUERY(sadr <* deriving_relationships | NOT (
        'AUTOMOTIVE_DESIGN.SYMMETRIC_SHAPE_ASPECT' IN TYPEOF(sadr.
        related_shape_aspect)))) = 0;
END_ENTITY;


ENTITY certification;
 name : label;
 purpose : text;
 kind : certification_type;
END_ENTITY;


ENTITY certification_assignment
 ABSTRACT SUPERTYPE;
 assigned_certification : certification;
 role : certification_role;
END_ENTITY;


ENTITY certification_role;
 name : label;
 description : text;
END_ENTITY;


ENTITY certification_type;
 description : label;
END_ENTITY;


ENTITY chamfer
 SUBTYPE OF (transition_feature);
 WHERE
  wr1 : SIZEOF(QUERY(fcr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr)) AND
         (fcr.name = 'first chamfer offset') AND (fcr.related_shape_aspect.
        description = 'chamfer offset') AND (
        'AUTOMOTIVE_DESIGN.CHAMFER_OFFSET' IN TYPEOF(fcr.related_shape_aspect)
        ))) = 1;
  wr2 : SIZEOF(QUERY(fcr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr)) AND
```

```
        (fcr.name = 'second chamfer offset') AND (fcr.related_shape_aspect.
       description IN ['chamfer offset','chamfer angle']) AND (
       'AUTOMOTIVE_DESIGN.CHAMFER_OFFSET' IN TYPEOF(fcr.related_shape_aspect)
       ))) = 1;
END_ENTITY;


ENTITY chamfer_offset
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : SELF.description IN ['chamfer offset','chamfer angle'];
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
       .used_representation))) = 1;
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
       'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
       .used_representation)) AND (SIZEOF(pdr.used_representation.items) = 1)
       )) = 1;
  wr4 : (SELF.description <> 'chamfer offset') OR (SIZEOF(QUERY(pdr <* USEDIN(
       SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
       'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
       .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
       used_representation.items | (srwp_i.name = 'offset amount') AND (
       SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
        = 1))) = 1);
  wr5 : (SELF.description = 'chamfer angle') OR (SIZEOF(QUERY(pdr <* USEDIN(
       SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF(QUERY(
       srwp_i <* pdr.used_representation.items | (srwp_i.name =
       'offset angle') AND (SIZEOF([
       'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) =
        2))) = 1))) = 1);
  wr6 : SIZEOF(QUERY(sar <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
       'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar))) >
       0;
  wr7 : (SELF.description = 'chamfer offset') OR (SIZEOF(QUERY(fcr <* USEDIN(
       SELF,'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
       'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr)) AND
        (fcr.name <> 'first chamfer offset') AND (fcr.name <>
       'second chamfer offset'))) = 0);
  wr8 : (SELF.description = 'chamfer angle') OR (SIZEOF(QUERY(fcr <* USEDIN(
       SELF,'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
       'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr)) AND
        (fcr.name <> 'second chamfer offset'))) = 0);
  wr9 : SIZEOF(QUERY(fcr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
```

```
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr)) AND
         ('AUTOMOTIVE_DESIGN.CHAMFER' IN TYPEOF(fcr.relating_shape_aspect)) OR
         (('AUTOMOTIVE_DESIGN.HOLE_BOTTOM' IN TYPEOF(fcr.relating_shape_aspect
        )) AND (fcr.relating_shape_aspect.description = 'flat with chamfer'))
        OR (('AUTOMOTIVE_DESIGN.SHAPE_ASPECT' IN TYPEOF(fcr.
        relating_shape_aspect)) AND (fcr.relating_shape_aspect.description =
        'top chamfer') AND ('AUTOMOTIVE_DESIGN.ROUND_HOLE' IN TYPEOF(fcr.
        relating_shape_aspect.of_shape.definition))))) = SIZEOF(QUERY(sar <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)));
END_ENTITY;


ENTITY characterized_object;
 name : label;
 description : text;
END_ENTITY;


ENTITY characterized_object_relationship;
 name : label;
 description : text;
 relating_object : characterized_object;
 related_object : characterized_object;
END_ENTITY;


ENTITY characterized_product_category
 SUBTYPE OF (characterized_object,product_category);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (pd.description <>
        'category attribute') OR (SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | (pdr.used_representation.description <> 'classification attribute'
        ) AND NOT ('AUTOMOTIVE_DESIGN.' +
        'EXTERNALLY_DEFINED_CLASSIFICATION_ATTRIBUTE' IN TYPEOF(pdr.
        used_representation)))) > 0))) = 0;
  wr2 : SIZEOF(QUERY(adr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DOCUMENT_REFERENCE.ITEMS') | adr.role.name =
        'classification source')) <= 1;
END_ENTITY;


ENTITY circle
 SUBTYPE OF (conic);
 radius : positive_length_measure;
END_ENTITY;


ENTITY circular_closed_profile
 SUBTYPE OF (shape_aspect);
```

```
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1;
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | SIZEOF(pdr.
        used_representation.items) = 2)) = 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | SIZEOF(QUERY(srwp_i
        <* pdr.used_representation.items | ('AUTOMOTIVE_DESIGN.PLACEMENT' IN
        TYPEOF(srwp_i)) AND (srwp_i.name = 'orientation'))) = 1)) = 1;
  wr5 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | SIZEOF(QUERY(srwp_i
        <* pdr.used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'diameter'))) = 1)) = 1;
END_ENTITY;


ENTITY circular_pattern
 SUBTYPE OF (replicate_feature);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND ({4<=SIZEOF(pdr.
        use_representation.items)<=5}))) = 1)) = 1;
  wr2 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | (r_i.name = 'diameter') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(r_i)) = 2))) =
        1))) = 1)) <= 1;
  wr3 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | (r_i.name = 'base feature rotation') AND (
        SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(r_i)) = 2)
        )) = 1))) = 1)) = 1;
  wr4 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.' +
```

```
        'DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | (
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (
        'AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN TYPEOF(r_i\
        measure_representation_item.value_component)) AND (r_i.name =
        'number of features'))) = 1))) = 1)) = 1;
  wr5 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.' +
        'DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | (r_i.name = 'angular spacing') AND (SIZEOF
        (['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(r_i)) = 2)
        )) = 1))) = 1)) = 1;
END_ENTITY;


ENTITY circular_runout_tolerance
 SUBTYPE OF (geometric_tolerance_with_datum_reference);
 WHERE
  wr1 : SIZEOF(SELF\geometric_tolerance_with_datum_reference.datum_system) <=
        2;
  wr2 : NOT ('AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT' IN
         TYPEOF(SELF));
END_ENTITY;


ENTITY circularity_tolerance
 SUBTYPE OF (geometric_tolerance);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE'
         IN TYPEOF(SELF));
  wr2 : NOT ('AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT' IN
         TYPEOF(SELF));
END_ENTITY;


ENTITY classification_system
 SUBTYPE OF (group);
 WHERE
  wr1 : (SIZEOF(USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.APPLIED_DOCUMENT_REFERENCE.ITEMS')) = 0) OR (SIZEOF
        (QUERY(adr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DOCUMENT_REFERENCE.ITEMS') | adr.role.name =
        'classification system source')) = 1);
END_ENTITY;


ENTITY closed_path_profile
 SUBTYPE OF (shape_aspect);
```

```
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) <= 1;
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(pdr.used_representation.items) <> 1
        ))) = 0;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.items | (
        'AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(pdr.used_representation.items)
        ) AND (srwp_i.name = 'orientation'))) = 1))) = 1;
  wr5 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.PATH_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation))) = 1;
END_ENTITY;


ENTITY closed_shell
 SUBTYPE OF (connected_face_set);
END_ENTITY;


ENTITY colour;
END_ENTITY;


ENTITY colour_rgb
 SUBTYPE OF (colour_specification);
 red : REAL;
 green : REAL;
 blue : REAL;
 WHERE
  wr1 : {0.0<=red<=1.0};
  wr2 : {0.0<=green<=1.0};
  wr3 : {0.0<=blue<=1.0};
END_ENTITY;


ENTITY colour_specification
 SUBTYPE OF (colour);
 name : label;
END_ENTITY;


ENTITY component_function_definition
 SUBTYPE OF (configuration_design);
```

```
 WHERE
  wr1 : SELF.design.frame_of_reference.name = 'functional definition';
  wr2 : 'AUTOMOTIVE_DESIGN.PRODUCT_COMPONENT' IN TYPEOF(SELF.configuration);
END_ENTITY;


ENTITY composite_curve
 SUBTYPE OF (bounded_curve);
 segments : LIST[1:?] OF composite_curve_segment;
 self_intersect : LOGICAL;
 DERIVE
  n_segments : INTEGER := SIZEOF(segments);
  closed_curve : LOGICAL := segments[n_segments].transition <> discontinuous;
 WHERE
  wr1 : ((NOT closed_curve) AND (SIZEOF(QUERY(temp <* segments | temp.
        transition = discontinuous)) = 1)) OR ((closed_curve) AND (SIZEOF(
        QUERY(temp <* segments | temp.transition = discontinuous)) = 0));
END_ENTITY;


ENTITY composite_curve_on_surface
 SUPERTYPE OF (boundary_curve)
 SUBTYPE OF (composite_curve);
 DERIVE
  basis_surface : SET[0:2] OF surface := get_basis_surface(SELF);
 WHERE
  wr1 : SIZEOF(basis_surface) > 0;
  wr2 : constraints_composite_curve_on_surface(SELF);
END_ENTITY;


ENTITY composite_curve_segment;
 transition : transition_code;
 same_sense : BOOLEAN;
 parent_curve : curve;
 INVERSE
  using_curves : BAG[1:?] OF composite_curve FOR segments;
 WHERE
  wr1 : ('AUTOMOTIVE_DESIGN.BOUNDED_CURVE' IN TYPEOF(parent_curve));
END_ENTITY;


ENTITY composite_shape_aspect
 SUBTYPE OF (shape_aspect);
 INVERSE
  component_relationships : SET[2:?] OF shape_aspect_relationship FOR
                            relating_shape_aspect;
END_ENTITY;


ENTITY composite_text
 SUBTYPE OF (geometric_representation_item);
 collected_text : SET[2:?] OF text_or_character;
 WHERE
```

```
  wr1 : acyclic_composite_text(SELF,SELF.collected_text);
END_ENTITY;


ENTITY composite_text_with_associated_curves
 SUBTYPE OF (composite_text);
 associated_curves : SET[1:?] OF curve;
END_ENTITY;


ENTITY composite_text_with_blanking_box
 SUBTYPE OF (composite_text);
 blanking : planar_box;
END_ENTITY;


ENTITY composite_text_with_extent
 SUBTYPE OF (composite_text);
 extent : planar_extent;
END_ENTITY;


ENTITY compound_feature_in_panel
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : (SIZEOF(['AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
        'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT'] * TYPEOF(SELF)) = 2) OR (
        SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN-PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | 'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT'
        IN TYPEOF(sa))) = 1))) = 1);
  wr2 : (('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND (
        SIZEOF(QUERY(fcr <* SELF\composite_shape_aspect.
        component_relationships |
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr))) =
        SIZEOF(SELF\composite_shape_aspect.component_relationships))) OR (NOT (
        'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND (
        SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(csa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT'
         IN TYPEOF(csa)) AND (SIZEOF(QUERY(fcr <* csa.component_relationships
         | 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr)))
         = SIZEOF(SELF\composite_shape_aspect.component_relationships)))) = 1)
        )) = 1));
  wr3 : (SIZEOF(['AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_SOLID',
        'AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
        'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT'] * TYPEOF(SELF)) <> 3) OR (
        SIZEOF(QUERY(sar <* SELF\composite_shape_aspect.
        component_relationships | SIZEOF([
        'AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
```

```
        'AUTOMOTIVE_DESIGN.TRANSITION_FEATURE'] * TYPEOF(sar.
        related_shape_aspect)) = 1)) = SIZEOF(SELF\composite_shape_aspect.
        component_relationships));
  wr4 : ('AUTOMOTIVE_DESIGN.INSTANCED_FEATURE' IN TYPEOF(SELF)) OR (SIZEOF(
        QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(csa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT'
         IN TYPEOF(csa)) AND (SIZEOF(QUERY(sar <* csa.component_relationships
         | SIZEOF(['AUTOMOTIVE_DESIGN.PLACED_FEATURE',
        'AUTOMOTIVE_DESIGN.TRANSITION_FEATURE'] * TYPEOF(sar.
        related_shape_aspect)) = 1)) <> SIZEOF(csa.component_relationships))))
         > 0))) = 0);
  wr5 : (('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND (
        SIZEOF(QUERY(fcr <* SELF\composite_shape_aspect.
        component_relationships | ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
        TYPEOF(fcr.related_shape_aspect)) AND (SIZEOF(QUERY(pdr <* USEDIN(fcr.
        related_shape_aspect,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(
        srwp_i)) AND (srwp_i.name = 'element placement'))) = 1))) = 1))) =
        SIZEOF(QUERY(fcr <* SELF\composite_shape_aspect.
        component_relationships | ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
        TYPEOF(fcr.related_shape_aspect)))))) OR (NOT (
        'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND (
        SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(csa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT'
         IN TYPEOF(csa)) AND (SIZEOF(QUERY(fcr <* csa.component_relationships
         | ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN TYPEOF(fcr.
        related_shape_aspect)) AND (SIZEOF(QUERY(pdr <* USEDIN(fcr.
        related_shape_aspect,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(
        srwp_i)) AND (srwp_i.name = 'element placement'))) = 1))) = 1))) =
        SIZEOF(QUERY(fcr <* SELF\composite_shape_aspect.
        component_relationships | ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
        TYPEOF(fcr.related_shape_aspect))))))) = 1))) = 1));
END_ENTITY;


ENTITY compound_feature_in_solid
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : (SIZEOF(['AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_SOLID',
        'AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
```

```
      'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT'] * TYPEOF(SELF)) = 3) OR (
      SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
      SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | 'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT'
      IN TYPEOF(sa))) = 1))) = 1);
wr2 : (('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND (
      SIZEOF(QUERY(fcr <* SELF\composite_shape_aspect.
      component_relationships |
      'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr))) =
      SIZEOF(SELF\composite_shape_aspect.component_relationships))) OR (NOT (
      'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND (
      SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
      SIZEOF(QUERY(csa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_SAPECT'
       IN TYPEOF(csa)) AND (SIZEOF(QUERY(fcr <* csa.component_relationships
       | 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr)))
       = SIZEOF(SELF\composite_shape_aspect.component_relationships)))) = 1)
      )) = 1));
wr3 : (SIZEOF(['AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_SOLID',
      'AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
      'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT'] * TYPEOF(SELF)) <> 3) OR (
      SIZEOF(QUERY(sar <* SELF\composite_shape_aspect.
      component_relationships | SIZEOF([
      'AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
      'AUTOMOTIVE_DESIGN.TRANSITION_FEATURE',
      'AUTOMOTIVE_DESIGN.THREAD_OCCURRENCE'] * TYPEOF(sar.
      related_shape_aspect)) = 1)) = SIZEOF(SELF\composite_shape_aspect.
      component_relationships));
wr4 : ('AUTOMOTIVE_DESIGN.INSTANCED_FEATURE' IN TYPEOF(SELF)) OR (SIZEOF(
      QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
      SIZEOF(QUERY(csa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | NOT (
      'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(csa)) OR (SIZEOF(
      QUERY(sar <* csa.component_relationships | SIZEOF([
      'AUTOMOTIVE_DESIGN.PLACED_FEATURE',
      'AUTOMOTIVE_DESIGN.TRANSITION_FEATURE',
      'AUTOMOTIVE_DESIGN.THREAD_OCCURRENCE'] * TYPEOF(sar.
      related_shape_aspect)) = 1)) <> SIZEOF(csa.component_relationships))))
       > 0))) = 0);
wr5 : (('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND (
      SIZEOF(QUERY(fcr <* SELF\composite_shape_aspect.
      component_relationships | ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
      TYPEOF(fcr.related_shape_aspect)) AND (SIZEOF(QUERY(pdr <* USEDIN(fcr.
      related_shape_aspect,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
```

```
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(
        srwp_i)) AND (srwp_i.name = 'element placement'))) = 1))) = 1))) =
        SIZEOF(QUERY(fcr <* SELF\composite_shape_aspect.
        component_relationships | ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
        TYPEOF(fcr.related_shape_aspect)))))) OR (NOT (
        'AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(SELF)) AND (
        SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(csa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | ('AUTOMOTIVE_DESIGN.COMPOSITE_SHAPE_ASPECT'
         IN TYPEOF(csa)) AND (SIZEOF(QUERY(fcr <* csa.component_relationships
         | ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN TYPEOF(fcr.
        related_shape_aspect)) AND (SIZEOF(QUERY(pdr <* USEDIN(fcr.
        related_shape_aspect,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(
        srwp_i)) AND (srwp_i.name = 'element placement'))) = 1))) = 1))) =
        SIZEOF(QUERY(fcr <* SELF\composite_shape_aspect.
        component_relationships | ('AUTOMOTIVE_DESIGN.PLACED_FEATURE' IN
        TYPEOF(fcr.related_shape_aspect))))))) = 1))) = 1));
END_ENTITY;


ENTITY compound_representation_item
 SUBTYPE OF (representation_item);
 item_element : compound_item_definition;
END_ENTITY;


ENTITY compound_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : ('AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_CONTEXT' IN TYPEOF(SELF.
        context_of_items)) AND (SELF.context_of_items\
        geometric_representation_context.coordinate_space_dimension = 3);
  wr2 : SIZEOF(QUERY(cbsr_i <* SELF.items | SIZEOF([
        'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_MODEL',
        'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM','AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D'
        ] * TYPEOF(cbsr_i)) <> 1)) = 0;
  wr3 : SIZEOF(QUERY(cbsr_i <* SELF.items | SIZEOF([
        'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_MODEL',
        'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM'] * TYPEOF(cbsr_i)) = 1)) > 0;
  wr4 : SIZEOF(QUERY(cbsr_i <* SELF.items | ('AUTOMOTIVE_DESIGN.MAPPED_ITEM'
        IN TYPEOF(cbsr_i)) AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.COMPOUND_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.NON_MANIFOLD_SURFACE_SHAPE_REPRESENTATION'] *
```

```
        TYPEOF(cbsr_i\mapped_item.mapping_source)) <> 1))) = 0;
END_ENTITY;


ENTITY concentricity_tolerance
 SUBTYPE OF (geometric_tolerance_with_datum_reference);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT' IN
         TYPEOF(SELF));
  wr2 : SIZEOF(SELF\geometric_tolerance_with_datum_reference.datum_system) = 1
        ;
END_ENTITY;


ENTITY concept_feature_relationship;
 name : label;
 description : text;
 relating_product_concept_feature : product_concept_feature;
 related_product_concept_feature : product_concept_feature;
END_ENTITY;


ENTITY concept_feature_relationship_with_condition
 SUBTYPE OF (concept_feature_relationship);
 conditional_operator : label;
END_ENTITY;


ENTITY conditional_concept_feature
 SUBTYPE OF (product_concept_feature);
 condition : concept_feature_relationship_with_condition;
END_ENTITY;


ENTITY configurable_item
 SUBTYPE OF (configuration_item);
 item_concept_feature : SET[1:?] OF product_concept_feature_association;
END_ENTITY;


ENTITY configuration_definition;
 pair_values : SET[1:?] OF pair_value;
 t_parameter : motion_parameter_measure;
END_ENTITY;


ENTITY configuration_design;
 name : label;
 description : text;
 configuration : configuration_item;
 design : configuration_design_item;
 UNIQUE
  ur1 : configuration,design;
END_ENTITY;
```

```
ENTITY configuration_interpolation;
 previous_configuration_definition : configuration_definition;
 next_configuration_definition : configuration_definition;
 interpolation : interpolation_type;
END_ENTITY;


ENTITY configuration_item;
 id : identifier;
 name : label;
 description : OPTIONAL text;
 item_concept : product_concept;
 purpose : OPTIONAL label;
 UNIQUE
  ur1 : id;
END_ENTITY;


ENTITY configuration_item_context
 SUBTYPE OF (application_context_element);
 usage : label;
END_ENTITY;


ENTITY configuration_item_relationship;
 name : label;
 description : text;
 relating_configuration_item : configuration_item;
 related_configuration_item : configuration_item;
END_ENTITY;


ENTITY configuration_item_with_context
 SUBTYPE OF (configuration_item);
 frame_of_reference : configuration_item_context;
END_ENTITY;


ENTITY configured_effective_context_assignment
 SUBTYPE OF (effective_context_assignment);
 items : SET[1:?] OF configured_effective_context_item;
 WHERE
  wr1 : SIZEOF(SELF.items) = 1;
  wr2 : SIZEOF(QUERY(ea <* USEDIN(SELF.assigned_effectivity,
        'AUTOMOTIVE_DESIGN.EFFECTIVITY_ASSIGNMENT.' + 'ASSIGNED_EFFECTIVITY')
         | 'AUTOMOTIVE_DESIGN.CONFIGURED_EFFECTIVE_ITEM_ASSIGNMENT' IN TYPEOF(
        ea))) = 1;
  wr3 : SELF.role.name = 'configuration';
END_ENTITY;


ENTITY configured_effective_item_assignment
 SUBTYPE OF (effective_item_assignment);
 items : SET[1:?] OF configured_effective_item;
```

```
 WHERE
  wr1 : SIZEOF(SELF.items) = 1;
  wr2 : SIZEOF(QUERY(ae <* USEDIN(SELF.assigned_effectivity,
        'AUTOMOTIVE_DESIGN.EFFECTIVITY_ASSIGNMENT.' + 'ASSIGNED_EFFECTIVITY')
         | 'AUTOMOTIVE_DESIGN.CONFIGURED_EFFECTIVE_CONTEXT_ASSIGNMENT' IN
        TYPEOF(ae))) = 1;
  wr3 : SELF.role.name = 'configuration';
END_ENTITY;


ENTITY conic
 SUPERTYPE OF (ONEOF(circle,ellipse,hyperbola,parabola))
 SUBTYPE OF (curve);
 position : axis2_placement;
END_ENTITY;


ENTITY conical_surface
 SUBTYPE OF (elementary_surface);
 radius : length_measure;
 semi_angle : plane_angle_measure;
 WHERE
  wr1 : radius >= 0.0;
END_ENTITY;


ENTITY connected_edge_set
 SUBTYPE OF (topological_representation_item);
 ces_edges : SET[1:?] OF edge;
END_ENTITY;


ENTITY connected_face_set
 SUPERTYPE OF (ONEOF(closed_shell,open_shell))
 SUBTYPE OF (topological_representation_item);
 cfs_faces : SET[1:?] OF face;
END_ENTITY;


ENTITY connected_face_shape_representation
 SUBTYPE OF (representation);
 WHERE
  wr1 : SIZEOF(SELF.items) = 1;
  wr2 : SIZEOF(QUERY(i <* SELF.items | ('AUTOMOTIVE_DESIGN.CLOSED_SHELL' IN
        TYPEOF(i)) OR ('AUTOMOTIVE_DESIGN.OPEN_SHELL' IN TYPEOF(i)))) = 1;
END_ENTITY;


ENTITY constructive_geometry_representation
 SUBTYPE OF (representation);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_CONTEXT' IN TYPEOF(SELF.
        context_of_items);
  wr2 : SIZEOF(QUERY(cgr_i <* SELF.items | SIZEOF([
```

```
        'AUTOMOTIVE_DESIGN.PLACEMENT','AUTOMOTIVE_DESIGN.CURVE',
        'AUTOMOTIVE_DESIGN.EDGE','AUTOMOTIVE_DESIGN.FACE',
        'AUTOMOTIVE_DESIGN.POINT','AUTOMOTIVE_DESIGN.SURFACE',
        'AUTOMOTIVE_DESIGN.VERTEX'] * TYPEOF(cgr_i)) <> 1)) = 0;
  wr3 : SIZEOF(QUERY(rr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_2') |
        'AUTOMTOIVE_DESIGN.CONSTRUCTIVE_GEOMETRY_REPRESENTATION_RELATIONSHIP'
        IN TYPEOF(rr))) < 1;
  wr4 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'REPRESENTATION_MAP.MAPPED_REPRESENTATION')) = 0;
END_ENTITY;


ENTITY constructive_geometry_representation_relationship
 SUBTYPE OF (representation_relationship);
 WHERE
  wr1 : SELF.rep_1.context_of_items :=: SELF.rep_2.context_of_items;
  wr2 : 'AUTOMOTIVE_DESIGN.CONSTRUCTIVE_GEOMETRY_REPRESENTATION' IN TYPEOF(
        SELF.rep_2);
  wr3 : SIZEOF(['AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.CONSTRUCTIVE_GEOMETRY_REPRESENTATION'] * TYPEOF(
        SELF.rep_1)) = 1;
  wr4 : NOT (
        'AUTOMOTIVE_DESIGN.REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION' IN
         TYPEOF(SELF));
END_ENTITY;


ENTITY context_dependent_invisibility
 SUBTYPE OF (invisibility);
 presentation_context : invisibility_context;
END_ENTITY;


ENTITY context_dependent_over_riding_styled_item
 SUBTYPE OF (over_riding_styled_item);
 style_context : SET[1:2] OF style_context_select;
 WHERE
  wr1 : (SIZEOF(QUERY(sc <* SELF.style_context |
        'AUTOMOTIVE_DESIGN.REPRESENTATION' IN TYPEOF(sc))) = 1) AND (SIZEOF(
        QUERY(sc <* SELF.style_context |
        'AUTOMOTIVE_DESIGN.REPRESENTATION_ITEM' IN TYPEOF(sc))) = 1);
END_ENTITY;


ENTITY context_dependent_shape_representation;
 name : label;
 description : text;
 representation_relation : shape_representation_relationship;
 represented_product_relation : product_definition_shape;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RELATIONSHIP' IN TYPEOF(SELF.
        represented_product_relation.definition);
END_ENTITY;
```

```
ENTITY context_dependent_unit
 SUBTYPE OF (named_unit);
 name : label;
END_ENTITY;


ENTITY contract;
 name : label;
 purpose : text;
 kind : contract_type;
END_ENTITY;


ENTITY contract_assignment
 ABSTRACT SUPERTYPE;
 assigned_contract : contract;
 role : contract_role;
END_ENTITY;


ENTITY contract_role;
 name : label;
 description : text;
END_ENTITY;


ENTITY contract_type;
 description : label;
END_ENTITY;


ENTITY conversion_based_unit
 SUBTYPE OF (named_unit);
 name : label;
 conversion_factor : measure_with_unit;
END_ENTITY;


ENTITY coordinated_universal_time_offset;
 hour_offset : hour_in_day;
 minute_offset : OPTIONAL minute_in_hour;
 sense : ahead_or_behind;
END_ENTITY;


ENTITY csg_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SELF\representation.context_of_items\geometric_representation_context.
        coordinate_space_dimension = 3;
  wr2 : SIZEOF(QUERY(it <* SELF\representation.items | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.CSG_SOLID','AUTOMOTIVE_DESIGN.EXTRUDED_FACE_SOLID',
        'AUTOMOTIVE_DESIGN.REVOLVED_FACE_SOLID',
```

```
        'AUTOMOTIVE_DESIGN.SOLID_REPLICA','AUTOMOTIVE_DESIGN.MAPPED_ITEM',
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D'] * TYPEOF(it)) = 1))) = 0;
  wr3 : SIZEOF(QUERY(it <* SELF\representation.items | SIZEOF([
        'AUTOMOTIVE_DESIGN.CSG_SOLID','AUTOMOTIVE_DESIGN.EXTRUDED_FACE_SOLID',
        'AUTOMOTIVE_DESIGN.REVOLVED_FACE_SOLID',
        'AUTOMOTIVE_DESIGN.SOLID_REPLICA','AUTOMOTIVE_DESIGN.MAPPED_ITEM'] *
        TYPEOF(it)) = 1)) >= 1;
  wr4 : SIZEOF(QUERY(it <* SELF\representation.items | (
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(it)) AND NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.CSG_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.FACETED_BREP_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.ADVANCED_BREP_SHAPE_REPRESENTATION'] * TYPEOF(it\
        mapped_item.mapping_source.mapped_representation)) > 0))) = 0;
END_ENTITY;


ENTITY csg_solid
 SUBTYPE OF (solid_model);
 tree_root_expression : csg_select;
END_ENTITY;


ENTITY curve
 SUPERTYPE OF (ONEOF(line,conic,pcurve,surface_curve,offset_curve_2d,
               offset_curve_3d,curve_replica))
 SUBTYPE OF (geometric_representation_item);
END_ENTITY;


ENTITY curve_bounded_surface
 SUBTYPE OF (bounded_surface);
 basis_surface : surface;
 boundaries : SET[1:?] OF boundary_curve;
 implicit_outer : BOOLEAN;
 WHERE
  wr1 : NOT (implicit_outer AND ('AUTOMOTIVE_DESIGN.OUTER_BOUNDARY_CURVE' IN
        TYPEOF(boundaries)));
  wr2 : (NOT (implicit_outer)) OR ('AUTOMOTIVE_DESIGN.BOUNDED_SURFACE' IN
        TYPEOF(basis_surface));
  wr3 : SIZEOF(QUERY(temp <* boundaries |
        'AUTOMOTIVE_DESIGN.OUTER_BOUNDARY_CURVE' IN TYPEOF(temp))) <= 1;
  wr4 : SIZEOF(QUERY(temp <* boundaries | (temp\composite_curve_on_surface.
        basis_surface[1] :<>: SELF.basis_surface))) = 0;
END_ENTITY;


ENTITY curve_dimension
 SUBTYPE OF (dimension_curve_directed_callout);
END_ENTITY;


ENTITY curve_replica
 SUBTYPE OF (curve);
 parent_curve : curve;
```

```
 transformation : cartesian_transformation_operator;
 WHERE
  wr1 : transformation.dim = parent_curve.dim;
  wr2 : acyclic_curve_replica(SELF,parent_curve);
END_ENTITY;


ENTITY curve_style;
 name : label;
 curve_font : curve_font_or_scaled_curve_font_select;
 curve_width : size_select;
 curve_colour : colour;
END_ENTITY;


ENTITY curve_style_font;
 name : label;
 pattern_list : LIST[1:?] OF curve_style_font_pattern;
END_ENTITY;


ENTITY curve_style_font_pattern;
 visible_segment_length : positive_length_measure;
 invisible_segment_length : positive_length_measure;
END_ENTITY;


ENTITY curve_style_rendering;
 rendering_method : shading_curve_method;
 rendering_properties : surface_rendering_properties;
END_ENTITY;


ENTITY cylindrical_pair
 SUBTYPE OF (kinematic_pair);
END_ENTITY;


ENTITY cylindrical_pair_range
 SUBTYPE OF (simple_pair_range);
 SELF\simple_pair_range.applies_to_pair : cylindrical_pair;
 lower_limit_actual_translation : translational_range_measure;
 upper_limit_actual_translation : translational_range_measure;
 lower_limit_actual_rotation : rotational_range_measure;
 upper_limit_actual_rotation : rotational_range_measure;
 WHERE
  wr1 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        lower_limit_actual_translation)) OR (
        'AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        upper_limit_actual_translation))) XOR (lower_limit_actual_translation
        < upper_limit_actual_translation);
  wr2 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        lower_limit_actual_rotation)) OR ('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE'
        IN TYPEOF(upper_limit_actual_rotation))) XOR (
```

```
        lower_limit_actual_rotation < upper_limit_actual_rotation);
END_ENTITY;


ENTITY cylindrical_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : cylindrical_pair;
 actual_translation : length_measure;
 actual_rotation : plane_angle_measure;
END_ENTITY;


ENTITY cylindrical_surface
 SUBTYPE OF (elementary_surface);
 radius : positive_length_measure;
END_ENTITY;


ENTITY cylindricity_tolerance
 SUBTYPE OF (geometric_tolerance);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE'
         IN TYPEOF(SELF));
  wr2 : NOT ('AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT' IN
         TYPEOF(SELF));
END_ENTITY;


ENTITY data_environment;
 name : label;
 description : text;
 elements : SET[1:?] OF property_definition_representation;
END_ENTITY;


ENTITY date
 SUPERTYPE OF (calendar_date);
 year_component : year_number;
END_ENTITY;


ENTITY date_and_time;
 date_component : date;
 time_component : local_time;
END_ENTITY;


ENTITY date_and_time_assignment
 ABSTRACT SUPERTYPE;
 assigned_date_and_time : date_and_time;
 role : date_time_role;
END_ENTITY;
```

```
ENTITY date_assignment
 ABSTRACT SUPERTYPE;
 assigned_date : date;
 role : date_role;
END_ENTITY;


ENTITY date_role;
 name : label;
 description : text;
END_ENTITY;


ENTITY date_time_role;
 name : label;
 description : text;
END_ENTITY;


ENTITY dated_effectivity
 SUBTYPE OF (effectivity);
 effectivity_start_date : date_and_time;
 effectivity_end_date : OPTIONAL date_and_time;
END_ENTITY;


ENTITY datum
 SUBTYPE OF (shape_aspect);
 identification : identifier;
 INVERSE
  established_by_relationships : SET[1:?] OF shape_aspect_relationship FOR
                                 related_shape_aspect;
 WHERE
  wr1 : SIZEOF(QUERY(x <* SELF.established_by_relationships | SIZEOF(TYPEOF(x.
        relating_shape_aspect) * ['AUTOMOTIVE_DESIGN.DATUM_FEATURE',
        'AUTOMOTIVE_DESIGN.DATUM_TARGET']) <> 1)) = 0;
END_ENTITY;


ENTITY datum_feature
 SUBTYPE OF (shape_aspect);
 INVERSE
  feature_basis_relationship : shape_aspect_relationship FOR
                               relating_shape_aspect;
 WHERE
  wr1 : SIZEOF(QUERY(sar <* bag_to_set(USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATING_SHAPE_ASPECT')) | NOT ('AUTOMOTIVE_DESIGN.DATUM' IN TYPEOF(
        sar.related_shape_aspect)))) = 0;
  wr2 : SELF.product_definitional = TRUE;
END_ENTITY;


ENTITY datum_feature_callout
```

```
 SUBTYPE OF (draughting_callout);
END_ENTITY;


ENTITY datum_reference;
 precedence : INTEGER;
 referenced_datum : datum;
 WHERE
  wr1 : precedence > 0;
END_ENTITY;


ENTITY datum_target
 SUBTYPE OF (shape_aspect);
 target_id : identifier;
 INVERSE
  target_basis_relationship : shape_aspect_relationship FOR
                              relating_shape_aspect;
 WHERE
  wr1 : SIZEOF(QUERY(sar <* bag_to_set(USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATING_SHAPE_ASPECT')) | NOT ('AUTOMOTIVE_DESIGN.DATUM' IN TYPEOF(
        sar.related_shape_aspect)))) = 0;
  wr2 : SELF.product_definitional = TRUE;
END_ENTITY;


ENTITY datum_target_callout
 SUBTYPE OF (draughting_callout);
END_ENTITY;


ENTITY default_tolerance_table
 SUBTYPE OF (representation);
 WHERE
  wr1 : SIZEOF(QUERY(rep_rel <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_1') |
        'AUTOMOTIVE_DESIGN.DEFAULT_TOLERANCE_TABLE_CELL' IN TYPEOF(rep_rel.
        rep_2))) > 0;
  wr2 : SELF\representation.context_of_items.context_type = 'default setting';
END_ENTITY;


ENTITY default_tolerance_table_cell
 SUBTYPE OF (representation);
 WHERE
  wr1 : SIZEOF(QUERY(rep_rel <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.REPRESENTATION_' + 'RELATIONSHIP.REP_2') |
        'AUTOMOTIVE_DESIGN.DEFAULT_TOLERANCE_TABLE' IN TYPEOF(rep_rel.rep_1)))
         = 1;
  wr2 : SELF\representation.context_of_items.context_type = 'default setting';
  wr3 : ((SIZEOF(QUERY(item <* SELF\representation.items | (
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' + 'ITEM' IN TYPEOF(item))
        AND (item\representation_item.name = 'significant number of digits')))
```

```
         = 1) AND (SIZEOF(QUERY(item <* SELF\representation.items | (
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' + 'ITEM' IN TYPEOF(item))
        AND (SIZEOF([item\representation_item.name] * ['lower limit',
        'upper limit']) > 0))) = 0)) OR ((SIZEOF(QUERY(item <* SELF\
        representation.items | ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
        'ITEM' IN TYPEOF(item)) AND (item\representation_item.name =
        'significant number of digits'))) = 0) AND (SIZEOF(QUERY(item <* SELF\
        representation.items | ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
        'ITEM' IN TYPEOF(item)) AND (item\representation_item.name =
        'lower limit'))) = 1) AND (SIZEOF(QUERY(item <* SELF\representation.
        items | ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' + 'ITEM' IN
        TYPEOF(item)) AND (item\representation_item.name = 'upper limit'))) =
        1));
  wr4 : ((SIZEOF(QUERY(item <* SELF\representation.items | (
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' + 'ITEM' IN TYPEOF(item))
        AND (item\representation_item.name = 'plus minus tolerance value'))) =
         1) AND (SIZEOF(QUERY(item <* SELF\representation.items | (
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' + 'ITEM' IN TYPEOF(item))
        AND (SIZEOF([item\representation_item.name] * ['lower tolerance value'
        ,'upper tolerance value']) > 0))) = 0)) OR ((SIZEOF(QUERY(item <* SELF
        \representation.items | ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
         'ITEM' IN TYPEOF(item)) AND (item\representation_item.name =
        'plus minus tolerance value'))) = 0) AND (SIZEOF(QUERY(item <* SELF\
        representation.items | ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
        'ITEM' IN TYPEOF(item)) AND (item\representation_item.name =
        'lower tolerance value'))) = 1) AND (SIZEOF(QUERY(item <* SELF\
        representation.items | ('AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_' +
        'ITEM' IN TYPEOF(item)) AND (item\representation_item.name =
        'upper tolerance value'))) = 1));
  wr5 : SIZEOF(SELF.items) <= 4;
END_ENTITY;


ENTITY defined_symbol
 SUBTYPE OF (geometric_representation_item);
 definition : defined_symbol_select;
 target : symbol_target;
END_ENTITY;


ENTITY definitional_representation
 SUBTYPE OF (representation);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.PARAMETRIC_REPRESENTATION_CONTEXT' IN TYPEOF(SELF\
        representation.context_of_items);
END_ENTITY;


ENTITY degenerate_pcurve
 SUBTYPE OF (point);
 basis_surface : surface;
 reference_to_curve : definitional_representation;
 WHERE
```

```
  wr1 : SIZEOF(reference_to_curve\representation.items) = 1;
  wr2 : 'AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF(reference_to_curve\representation.
        items[1]);
  wr3 : reference_to_curve\representation.items[1]\
        geometric_representation_item.dim = 2;
END_ENTITY;


ENTITY degenerate_toroidal_surface
 SUBTYPE OF (toroidal_surface);
 select_outer : BOOLEAN;
 WHERE
  wr1 : major_radius < minor_radius;
END_ENTITY;


ENTITY derived_shape_aspect
 SUPERTYPE OF (ONEOF(apex,centre_of_symmetry,geometric_alignment,
               geometric_intersection,parallel_offset,perpendicular_to,
               extension,tangent))
 SUBTYPE OF (shape_aspect);
 INVERSE
  deriving_relationships : SET[1:?] OF shape_aspect_deriving_relationship FOR
                           relating_shape_aspect;
END_ENTITY;


ENTITY derived_unit;
 elements : SET[1:?] OF derived_unit_element;
 WHERE
  wr1 : (SIZEOF(elements) > 1) OR ((SIZEOF(elements) = 1) AND (elements[1].
        exponent <> 1.0));
END_ENTITY;


ENTITY derived_unit_element;
 unit : named_unit;
 exponent : REAL;
END_ENTITY;


ENTITY descriptive_representation_item
 SUBTYPE OF (representation_item);
 description : text;
END_ENTITY;


ENTITY diameter_dimension
 SUBTYPE OF (dimension_curve_directed_callout);
END_ENTITY;


ENTITY digital_document_reference
 SUBTYPE OF (document_reference);
 items : SET[1:?] OF digital_document_reference_item;
```

```
 WHERE
  wr1 : SELF.assigned_document.kind.product_data_type = 'digital';
  wr2 : SELF.role.name = 'external definition';
  wr3 : SIZEOF(SELF.items) = 1;
END_ENTITY;


ENTITY dimension_callout_component_relationship
 SUBTYPE OF (draughting_callout_relationship);
 WHERE
  wr1 : SELF.name IN ['prefix','suffix'];
  wr2 : 'AUTOMOTIVE_DESIGN.STRUCTURED_DIMENSION_CALLOUT' IN TYPEOF(SELF.
        relating_draughting_callout);
  wr3 : SIZEOF(TYPEOF(SELF.related_draughting_callout) * [
        'AUTOMOTIVE_DESIGN.LEADER_DIRECTED_CALLOUT',
        'AUTOMOTIVE_DESIGN.PROJECTION_DIRECTED_CALLOUT',
        'AUTOMOTIVE_DESIGN.DIMENSION_CURVE_DIRECTED_CALLOUT',
        'AUTOMOTIVE_DESIGN.STRUCTURED_DIMENSION_CALLOUT']) = 0;
  wr4 : SELF.related_draughting_callout.contents * SELF.
        relating_draughting_callout.contents = SELF.related_draughting_callout
        .contents;
  wr5 : ((SELF.name = 'prefix') AND (SIZEOF(QUERY(ato <* QUERY(con <* SELF.
        related_draughting_callout.contents | (
        'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(con))) | NOT (
        ato.name = 'prefix text'))) = 0));
  wr6 : ((SELF.name = 'suffix') AND (SIZEOF(QUERY(ato <* QUERY(con <* SELF.
        related_draughting_callout.contents | (
        'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(con))) | NOT (
        ato.name = 'suffix text'))) = 0));
END_ENTITY;


ENTITY dimension_callout_relationship
 SUBTYPE OF (draughting_callout_relationship);
 WHERE
  wr1 : SELF.name IN ['primary','secondary'];
  wr2 : SIZEOF(TYPEOF(SELF.relating_draughting_callout) * [
        'AUTOMOTIVE_DESIGN.ANGULAR_DIMENSION',
        'AUTOMOTIVE_DESIGN.CURVE_DIMENSION',
        'AUTOMOTIVE_DESIGN.DIAMETER_DIMENSION',
        'AUTOMOTIVE_DESIGN.LEADER_DIRECTED_DIMENSION',
        'AUTOMOTIVE_DESIGN.LINEAR_DIMENSION',
        'AUTOMOTIVE_DESIGN.ORDINATE_DIMENSION',
        'AUTOMOTIVE_DESIGN.RADIUS_DIMENSION']) = 1;
  wr3 : SIZEOF(TYPEOF(SELF.related_draughting_callout) * [
        'AUTOMOTIVE_DESIGN.DIMENSION_CURVE_DIRECTED_CALLOUT',
        'AUTOMOTIVE_DESIGN.PROJECTION_DIRECTED_CALLOUT',
        'AUTOMOTIVE_DESIGN.LEADER_DIRECTED_CALLOUT']) = 0;
  wr4 : SELF.related_draughting_callout.contents * SELF.
        relating_draughting_callout.contents = SELF.related_draughting_callout
        .contents;
END_ENTITY;
```

```
ENTITY dimension_curve
 SUBTYPE OF (annotation_curve_occurrence);
 WHERE
  wr1 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'DIMENSION_CURVE_TERMINATOR.ANNOTATED_DIMENSION_' + 'CURVE')) <= 2;
  wr2 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'DIMENSION_CURVE_DIRECTED_CALLOUT.CONTENTS')) >= 1;
  wr3 : (SIZEOF(QUERY(dct_1 <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'DIMENSION_CURVE_TERMINATOR.ANNOTATED_DIMENSION_' + 'CURVE') | (dct_1.
        role = origin))) <= 1) AND (SIZEOF(QUERY(dct_2 <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.DIMENSION_CURVE_TERMINATOR.' +
        'ANNOTATED_DIMENSION_CURVE') | (dct_2.role = target))) <= 1);
END_ENTITY;


ENTITY dimension_curve_directed_callout
 SUBTYPE OF (draughting_callout);
 WHERE
  wr1 : SIZEOF(QUERY(d_c <* SELF\draughting_callout.contents |
        'AUTOMOTIVE_DESIGN.DIMENSION_CURVE' IN (TYPEOF(d_c)))) = 1;
  wr2 : SIZEOF(SELF\draughting_callout.contents) >= 2;
END_ENTITY;


ENTITY dimension_curve_terminator
 SUBTYPE OF (terminator_symbol);
 role : dimension_extent_usage;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.DIMENSION_CURVE' IN TYPEOF(SELF\terminator_symbol.
        annotated_curve);
END_ENTITY;


ENTITY dimension_pair
 SUBTYPE OF (draughting_callout_relationship);
 WHERE
  wr1 : SELF.name IN ['chained','parallel'];
  wr2 : SIZEOF(TYPEOF(SELF.relating_draughting_callout) * [
        'AUTOMOTIVE_DESIGN.ANGULAR_DIMENSION',
        'AUTOMOTIVE_DESIGN.CURVE_DIMENSION',
        'AUTOMOTIVE_DESIGN.DIAMETER_DIMENSION',
        'AUTOMOTIVE_DESIGN.LINEAR_DIMENSION',
        'AUTOMOTIVE_DESIGN.ORDINATE_DIMENSION',
        'AUTOMOTIVE_DESIGN.RADIUS_DIMENSION']) = 1;
  wr3 : SIZEOF(TYPEOF(SELF.related_draughting_callout) * [
        'AUTOMOTIVE_DESIGN.ANGULAR_DIMENSION',
        'AUTOMOTIVE_DESIGN.CURVE_DIMENSION',
        'AUTOMOTIVE_DESIGN.DIAMETER_DIMENSION',
        'AUTOMOTIVE_DESIGN.LINEAR_DIMENSION',
        'AUTOMOTIVE_DESIGN.ORDINATE_DIMENSION',
        'AUTOMOTIVE_DESIGN.RADIUS_DIMENSION']) = 1;
END_ENTITY;
```

```
ENTITY dimension_related_tolerance_zone_element;
 related_dimension : dimensional_location;
 related_element : tolerance_zone_definition;
END_ENTITY;


ENTITY dimension_text_associativity
 SUBTYPE OF (text_literal,mapped_item);
 WHERE
  wr1 : ('AUTOMOTIVE_DESIGN.SHAPE_DIMENSION_REPRESENTATION' IN TYPEOF(SELF\
        mapped_item.mapping_source.mapped_representation));
  wr2 : SIZEOF(SELF\mapped_item.mapping_source.mapped_representation.items) =
        1;
  wr3 : ('AUTOMOTIVE_DESIGN.DRAUGHTING_CALLOUT' IN TYPEOF(SELF\mapped_item.
        mapping_target));
  wr4 : SIZEOF(QUERY(ato <* QUERY(si <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.STYLED_ITEM.ITEM') | (
        'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(si))) | NOT (
        SIZEOF(QUERY(dc <* USEDIN(ato,'AUTOMOTIVE_DESIGN.DRAUGHTING_CALLOUT.'
        + 'CONTENTS') | ('AUTOMOTIVE_DESIGN.DRAUGHTING_CALLOUT' IN TYPEOF(dc))
        ) * [SELF\mapped_item.mapping_target]) = 1))) = 0;
END_ENTITY;


ENTITY dimensional_characteristic_representation;
 dimension : dimensional_characteristic;
 representation : shape_dimension_representation;
END_ENTITY;


ENTITY dimensional_exponents;
 length_exponent : REAL;
 mass_exponent : REAL;
 time_exponent : REAL;
 electric_current_exponent : REAL;
 thermodynamic_temperature_exponent : REAL;
 amount_of_substance_exponent : REAL;
 luminous_intensity_exponent : REAL;
END_ENTITY;


ENTITY dimensional_location
 SUPERTYPE OF (ONEOF(angular_location,dimensional_location_with_path))
 SUBTYPE OF (shape_aspect_relationship);
END_ENTITY;


ENTITY dimensional_location_with_path
 SUBTYPE OF (dimensional_location);
 path : shape_aspect;
END_ENTITY;


ENTITY dimensional_size
```

```
 SUPERTYPE OF (angular_size);
 applies_to : shape_aspect;
 name : label;
 WHERE
  wr1 : applies_to.product_definitional = TRUE;
END_ENTITY;


ENTITY directed_action
 SUBTYPE OF (executed_action);
 directive : action_directive;
END_ENTITY;


ENTITY directed_angle
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1;
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(pdr.used_representation.items) = 2)
        )) = 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i))
         AND (srwp_i.name = 'angle reference') AND (srwp_i.description IN [
        'punch direction','profile normal','surface normal']))) = 1))) = 1;
  wr5 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'angle') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) =
         2))) = 1))) = 1;
END_ENTITY;


ENTITY directed_dimensional_location
 SUBTYPE OF (dimensional_location);
END_ENTITY;


ENTITY direction
 SUBTYPE OF (geometric_representation_item);
```

```
 direction_ratios : LIST[2:3] OF REAL;
 WHERE
  wr1 : SIZEOF(QUERY(tmp <* direction_ratios | tmp <> 0.0)) > 0;
END_ENTITY;

ENTITY direction_shape_representation
 SUBTYPE OF (representation);
 WHERE
  wr1 : SIZEOF(SELF.items) = 1;
  wr2 : SIZEOF(QUERY(i <* SELF.items | 'AUTOMOTIVE_DESIGN.DIRECTION' IN TYPEOF
        (i))) = 1;
END_ENTITY;

ENTITY document;
 id : identifier;
 name : label;
 description : text;
 kind : document_type;
END_ENTITY;

ENTITY document_reference
 ABSTRACT SUPERTYPE;
 assigned_document : document;
 source : label;
 role : document_role;
END_ENTITY;

ENTITY document_relationship;
 name : label;
 description : text;
 relating_document : document;
 related_document : document;
END_ENTITY;

ENTITY document_role;
 name : label;
 description : text;
END_ENTITY;

ENTITY document_type;
 product_data_type : label;
END_ENTITY;

ENTITY document_usage_constraint;
 source : document;
 subject_element : label;
 subject_element_value : text;
END_ENTITY;
```

```
ENTITY draughting_annotation_occurrence
 SUBTYPE OF (annotation_occurrence);
 WHERE
  wr1 : (NOT ('AUTOMOTIVE_DESIGN.ANNOTATION_CURVE_OCCURRENCE' IN TYPEOF(SELF))
        ) OR (SIZEOF(QUERY(sty <* SELF.styles | NOT ((SIZEOF(sty.styles) = 1)
        AND ('AUTOMOTIVE_DESIGN.CURVE_STYLE' IN TYPEOF(sty.styles[1])))))) = 0)
        ;
  wr2 : (NOT ('AUTOMOTIVE_DESIGN.ANNOTATION_FILL_AREA_OCCURRENCE' IN TYPEOF(
        SELF))) OR (SIZEOF(QUERY(sty <* SELF.styles | NOT ((SIZEOF(sty.styles)
         = 1) AND ('AUTOMOTIVE_DESIGN.FILL_AREA_STYLE' IN TYPEOF(sty.styles[1]
        )))))) = 0);
  wr3 : (NOT ('AUTOMOTIVE_DESIGN.ANNOTATION_FILL_AREA_OCCURRENCE' IN TYPEOF(
        SELF))) OR (SIZEOF(QUERY(bound <* SELF.item\annotation_fill_area.
        boundaries | NOT (SIZEOF(QUERY(si <* USEDIN(bound,'AUTOMOTIVE_DESIGN.'
         + 'STYLED_ITEM.ITEM') | ('AUTOMOTIVE_DESIGN.' +
        'ANNOTATION_CURVE_OCCURRENCE' IN TYPEOF(si)))) > 0))) = 0);
  wr4 : (NOT ('AUTOMOTIVE_DESIGN.ANNOTATION_SYMBOL_OCCURRENCE' IN TYPEOF(SELF)
        )) OR (SIZEOF(QUERY(sty <* SELF.styles | NOT ((SIZEOF(sty.styles) = 1)
         AND (SIZEOF(TYPEOF(sty.styles[1]) * ['AUTOMOTIVE_DESIGN.SYMBOL_STYLE'
        ,'AUTOMOTIVE_DESIGN.NULL_STYLE']) = 1)))) = 0);
  wr5 : (NOT (('AUTOMOTIVE_DESIGN.ANNOTATION_SYMBOL_OCCURRENCE' IN TYPEOF(SELF
        )) AND ('AUTOMOTIVE_DESIGN.ANNOTATION_SYMBOL' IN TYPEOF(SELF.item))))
        OR (SIZEOF(['AUTOMOTIVE_DESIGN.' + 'DRAUGHTING_SYMBOL_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.' + 'DRAUGHTING_SUBFIGURE_REPRESENTATION'] * TYPEOF
        (SELF.item\mapped_item.mapping_source.mapped_representation)) = 1);
  wr6 : (NOT ('AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(SELF)))
         OR (SIZEOF(QUERY(sty <* SELF.styles | NOT ((SIZEOF(sty.styles) = 1)
        AND ('AUTOMOTIVE_DESIGN.TEXT_STYLE' IN TYPEOF(sty.styles[1])))))) = 0);
  wr7 : NOT (('AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(SELF))
        AND (SIZEOF(TYPEOF(SELF.item) * ['AUTOMOTIVE_DESIGN.COMPOSITE_TEXT',
        'AUTOMOTIVE_DESIGN.TEXT_LITERAL']) = 1));
  wr8 : (NOT (('AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(SELF))
         AND ('AUTOMOTIVE_DESIGN.COMPOSITE_TEXT' IN TYPEOF(SELF.item)))) OR (
        SIZEOF(QUERY(tl <* SELF.item\composite_text.collected_text | NOT (
        'AUTOMOTIVE_DESIGN.TEXT_LITERAL' IN TYPEOF(tl)))) = 0);
  wr9 : (NOT (('AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(SELF))
         AND ('AUTOMOTIVE_DESIGN.TEXT_LITERAL' IN TYPEOF(SELF.item)))) OR (
        SELF.item\text_literal.alignment IN ['baseline left','baseline centre'
        ,'baseline right']);
  wr10 : (NOT (('AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(SELF)
         ) AND ('AUTOMOTIVE_DESIGN.COMPOSITE_TEXT' IN TYPEOF(SELF.item)))) OR
         (SIZEOF(QUERY(tl <* QUERY(text <* SELF.item\composite_text.
         collected_text | ('AUTOMOTIVE_DESIGN.TEXT_LITERAL' IN TYPEOF(text)))
          | NOT (tl\text_literal.alignment IN ['baseline left',
         'baseline centre','baseline right'])))) = 0);
  wr11 : NOT ((('AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(SELF)
         ) AND ('AUTOMOTIVE_DESIGN.TEXT_LITERAL' IN TYPEOF(SELF.item))) AND (
         EXISTS(SELF.item\text_literal.placement.ref_direction)));
  wr12 : (NOT (('AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(SELF)
         ) AND ('AUTOMOTIVE_DESIGN.COMPOSITE_TEXT' IN TYPEOF(SELF.item)))) OR
         (SIZEOF(QUERY(tl <* QUERY(text <* SELF.item\composite_text.
         collected_text | ('AUTOMOTIVE_DESIGN.TEXT_LITERAL' IN TYPEOF(text)))
```

```
        | EXISTS(tl\text_literal.placement.ref_direction))) = 0);
wr13 : NOT (('AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(SELF))
        AND ('AUTOMOTIVE_DESIGN.COMPOSITE_TEXT' IN TYPEOF(SELF.item))) OR
       check_text_alignment(SELF.item);
wr14 : NOT (('AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(SELF))
        AND ('AUTOMOTIVE_DESIGN.COMPOSITE_TEXT' IN TYPEOF(SELF.item))) OR
       check_text_font(SELF.item);
wr15 : (NOT (('AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(SELF)
       ) AND ('AUTOMOTIVE_DESIGN.COMPOSITE_TEXT' IN TYPEOF(SELF.item)))) OR
       (SIZEOF(QUERY(tl <* QUERY(text <* SELF.item\composite_text.
       collected_text | ('AUTOMOTIVE_DESIGN.TEXT_LITERAL' IN TYPEOF(text)))
        | NOT (SIZEOF(TYPEOF(tl) * ['AUTOMOTIVE_DESIGN.' +
       'TEXT_LITERAL_WITH_BLANKING_BOX','AUTOMOTIVE_DESIGN.' +
       'TEXT_LITERAL_WITH_ASSOCIATED_CURVES']) = 0))) = 0);
wr16 : (NOT (('AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(SELF)
       ) AND ('AUTOMOTIVE_DESIGN.TEXT_LITERAL_WITH_ASSOCIATED_CURVES' IN
       TYPEOF(SELF.item)))) OR (SIZEOF(QUERY(crv <* SELF.item\
       text_literal_with_associated_curves.associated_curves | NOT (SIZEOF(
       QUERY(si <* USEDIN(crv,'AUTOMOTIVE_DESIGN.' + 'STYLED_ITEM.ITEM') | (
       'AUTOMOTIVE_DESIGN.ANNOTATION_CURVE_OCCURRENCE' IN TYPEOF(si)))) > 0)
       )) = 0);
wr17 : (NOT (('AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(SELF)
       ) AND ('AUTOMOTIVE_DESIGN.COMPOSITE_TEXT_WITH_ASSOCIATED_CURVES' IN
       TYPEOF(SELF.item)))) OR (SIZEOF(QUERY(crv <* SELF.item\
       composite_text_with_associated_curves.associated_curves | NOT (SIZEOF
       (QUERY(si <* USEDIN(crv,'AUTOMOTIVE_DESIGN.' + 'STYLED_ITEM.ITEM') |
       ('AUTOMOTIVE_DESIGN.ANNOTATION_CURVE_OCCURRENCE' IN TYPEOF(si)))) > 0
       ))) = 0);
wr18 : SIZEOF(QUERY(cs <* QUERY(sty <* SELF.styles | (
       'AUTOMOTIVE_DESIGN.CURVE_STYLE' IN TYPEOF(sty.styles[1]))) | NOT ((
       'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT' IN TYPEOF(cs.styles[1]\
       curve_style.curve_width)) AND (
       'AUTOMOTIVE_DESIGN.POSITIVE_LENGTH_MEASURE' IN TYPEOF(cs.styles[1]\
       curve_style.curve_width\measure_with_unit.value_component))))) = 0;
wr19 : SIZEOF(QUERY(fas <* QUERY(sty <* SELF.styles | (
       'AUTOMOTIVE_DESIGN.FILL_AREA_STYLE' IN TYPEOF(sty.styles[1]))) | NOT (
       (SIZEOF(QUERY(fs <* fas.styles[1]\fill_area_style.fill_styles | (
       'AUTOMOTIVE_DESIGN.FILL_AREA_STYLE_TILES' IN TYPEOF(fs)))) <= 1) AND
       (SIZEOF(QUERY(fst <* QUERY(fs <* fas.styles[1]\fill_area_style.
       fill_styles | ('AUTOMOTIVE_DESIGN.FILL_AREA_STYLE_TILES' IN TYPEOF(fs
       ))) | NOT (SIZEOF(fst\fill_area_style_tiles.tiles) = 1))) = 0)))) = 0
       ;
wr20 : SIZEOF(QUERY(fas <* QUERY(sty <* SELF.styles | (
       'AUTOMOTIVE_DESIGN.FILL_AREA_STYLE' IN TYPEOF(sty.styles[1]))) | NOT (
       SIZEOF(QUERY(fsh <* QUERY(fs <* fas.styles[1]\fill_area_style.
       fill_styles | ('AUTOMOTIVE_DESIGN.FILL_AREA_STYLE_HATCHING' IN TYPEOF
       (fs))) | NOT (fsh\fill_area_style_hatching.
       point_of_reference_hatch_line :=: fsh\fill_area_style_hatching.
       pattern_start))) = 0))) = 0;
wr21 : SIZEOF(QUERY(ts <* QUERY(sty <* SELF.styles | (
       'AUTOMOTIVE_DESIGN.TEXT_STYLE' IN TYPEOF(sty.styles[1]))) | NOT (
       'AUTOMOTIVE_DESIGN.TEXT_STYLE_WITH_BOX_CHARACTERISTICS' IN TYPEOF(ts.
```

```
        styles[1]))))) = 0;
  wr22 : SIZEOF(QUERY(ts <* QUERY(sty <* SELF.styles | (
        'AUTOMOTIVE_DESIGN.TEXT_STYLE_WITH_BOX_CHARACTERISTICS' IN TYPEOF(sty
        .styles[1]))) | NOT (SIZEOF(ts.styles[1]\
        text_style_with_box_characteristics.characteristics) = 4))) = 0;
END_ENTITY;


ENTITY draughting_callout
 SUBTYPE OF (geometric_representation_item);
 contents : SET[1:?] OF draughting_callout_element;
END_ENTITY;


ENTITY draughting_callout_relationship;
 name : label;
 description : text;
 relating_draughting_callout : draughting_callout;
 related_draughting_callout : draughting_callout;
END_ENTITY;


ENTITY draughting_elements
 SUBTYPE OF (draughting_callout);
 WHERE
  wr1 : SIZEOF(QUERY(l_c <* QUERY(con <* SELF.contents | (
        'AUTOMOTIVE_DESIGN.LEADER_CURVE' IN TYPEOF(con))) | NOT (SIZEOF(QUERY(
        ldc <* USEDIN(l_c,'AUTOMOTIVE_DESIGN.' + 'DRAUGHTING_CALLOUT.CONTENTS'
        ) | ('AUTOMOTIVE_DESIGN.LEADER_DIRECTED_CALLOUT' IN TYPEOF(ldc)))) <=
        1))) = 0;
  wr2 : NOT ('AUTOMOTIVE_DESIGN.DIMENSION_CURVE_DIRECTED_CALLOUT' IN TYPEOF(
        SELF)) OR (SIZEOF(QUERY(con <* SELF.contents | (
        'AUTOMOTIVE_DESIGN.PROJECTION_CURVE' IN TYPEOF(con)))) <= 2);
  wr3 : SIZEOF(QUERY(rc <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.DIMENSION_CALLOUT_'
        + 'RELATIONSHIP.RELATING_DRAUGHTING_CALLOUT') | rc.name = 'primary'))
        <= 1;
  wr4 : SIZEOF(QUERY(rc <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.DIMENSION_CALLOUT_'
        + 'RELATIONSHIP.RELATING_DRAUGHTING_CALLOUT') | rc.name = 'secondary')
        ) <= 1;
  wr5 : SIZEOF(QUERY(sec <* QUERY(rc <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.DIMENSION_CALLOUT_' +
        'RELATIONSHIP.RELATING_DRAUGHTING_CALLOUT') | rc.name = 'secondary')
         | NOT (SIZEOF(QUERY(prim <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.DIMENSION_CALLOUT_' +
        'RELATIONSHIP.RELATING_DRAUGHTING_CALLOUT') | prim.name = 'primary'))
        = 1))) = 0;
END_ENTITY;


ENTITY draughting_model
 SUBTYPE OF (representation);
 UNIQUE
  ur1 : SELF\representation.name;
 WHERE
```

```
  wr1 : SIZEOF(QUERY(it <* SELF.items | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM','AUTOMOTIVE_DESIGN.STYLED_ITEM',
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT',
        'AUTOMOTIVE_DESIGN.CAMERA_MODEL_D3'] * TYPEOF(it)) = 1))) = 0;
  wr2 : SIZEOF(QUERY(mi <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(it))) | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.DRAUGHTING_MODEL'] * TYPEOF(mi\mapped_item.
        mapping_source.mapped_representation)) = 1))) = 0;
  wr3 : SIZEOF(QUERY(smi <* QUERY(si <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it))) | (
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(si\styled_item.item))) | NOT
        (('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN TYPEOF(smi\styled_item.
        item\mapped_item.mapping_source.mapped_representation)) AND (SIZEOF(
        QUERY(sty <* smi\styled_item.styles | NOT (SIZEOF(QUERY(psa <* sty.
        styles | NOT ('AUTOMOTIVE_DESIGN.CURVE_STYLE' IN TYPEOF(psa)))) = 1)))
         = 1)))) = 0;
END_ENTITY;


ENTITY draughting_pre_defined_colour
 SUBTYPE OF (pre_defined_colour);
 WHERE
  wr1 : SELF.name IN ['red','green','blue','yellow','magenta','cyan','black',
        'white'];
END_ENTITY;


ENTITY draughting_pre_defined_curve_font
 SUBTYPE OF (pre_defined_curve_font);
 WHERE
  wr1 : SELF.name IN ['continuous','chain','chain double dash','dashed',
        'dotted'];
END_ENTITY;


ENTITY draughting_pre_defined_text_font
 SUBTYPE OF (pre_defined_text_font);
 WHERE
  wr1 : SELF.name IN ['ISO 3098-1 font A','ISO 3098-1 font B'];
END_ENTITY;


ENTITY draughting_specification_reference
 SUBTYPE OF (document_reference);
 specified_items : SET[1:?] OF specified_item;
 WHERE
  wr1 : (SELF.assigned_document.kind.product_data_type =
        'draughting specification');
END_ENTITY;


ENTITY draughting_subfigure_representation
 SUBTYPE OF (symbol_representation);
```

```
 WHERE
  wr1 : SIZEOF(QUERY(item <* SELF\representation.items | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.DRAUGHTING_CALLOUT',
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT'] * TYPEOF(item)) = 1))) = 0;
  wr2 : SIZEOF(QUERY(item <* SELF\representation.items | SIZEOF([
        'AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.DRAUGHTING_CALLOUT'] * TYPEOF(item)) = 1)) >= 1;
  wr3 : SIZEOF(QUERY(srm <* QUERY(rm <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'REPRESENTATION_MAP.MAPPED_REPRESENTATION') | (
        'AUTOMOTIVE_DESIGN.SYMBOL_REPRESENTATION_MAP' IN TYPEOF(rm))) | NOT (
        SIZEOF(QUERY(a_s <* QUERY(mi <* srm.map_usage | (
        'AUTOMOTIVE_DESIGN.ANNOTATION_SYMBOL' IN TYPEOF(mi))) | NOT (SIZEOF(
        QUERY(aso <* USEDIN(a_s,'AUTOMOTIVE_DESIGN.STYLED_ITEM.ITEM') | NOT (
        'AUTOMOTIVE_DESIGN.ANNOTATION_SUBFIGURE_OCCURRENCE' IN TYPEOF(aso))))
        = 0))) = 0))) > 0;
  wr4 : NOT (acyclic_mapped_item_usage(SELF));
  wr5 : SIZEOF(SELF.context_of_items.representations_in_context) = 1;
END_ENTITY;


ENTITY draughting_symbol_representation
 SUBTYPE OF (symbol_representation);
 UNIQUE
  ur1 : SELF\representation.name;
 WHERE
  wr1 : SIZEOF(QUERY(item <* SELF\representation.items | NOT (SIZEOF(TYPEOF(
        item) * ['AUTOMOTIVE_DESIGN.ANNOTATION_CURVE_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.ANNOTATION_SYMBOL_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.ANNOTATION_FILL_AREA_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT']) = 1))) = 0;
  wr2 : SIZEOF(QUERY(item <* SELF\representation.items | (SIZEOF(TYPEOF(item)
        * ['AUTOMOTIVE_DESIGN.ANNOTATION_CURVE_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.ANNOTATION_SYMBOL_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.ANNOTATION_FILL_AREA_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE']) = 1))) >= 1;
  wr3 : SIZEOF(QUERY(item <* SELF\representation.items |
        'AUTOMOTIVE_DESIGN.ANNOTATION_SUBFIGURE_OCCURRENCE' IN TYPEOF(item)))
        = 0;
  wr4 : SIZEOF(QUERY(srm <* QUERY(rm <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'REPRESENTATION_MAP.MAPPED_REPRESENTATION') | (
        'AUTOMOTIVE_DESIGN.SYMBOL_REPRESENTATION_MAP' IN TYPEOF(rm))) | NOT (
        SIZEOF(QUERY(a_s <* QUERY(mi <* srm.map_usage | (
        'AUTOMOTIVE_DESIGN.ANNOTATION_SYMBOL' IN TYPEOF(mi))) | NOT (SIZEOF(
        QUERY(aso <* USEDIN(a_s,'AUTOMOTIVE_DESIGN.STYLED_ITEM.ITEM') | NOT (
        'AUTOMOTIVE_DESIGN.ANNOTATION_SYMBOL_OCCURRENCE' IN TYPEOF(aso)))) = 0
        ))) = 0))) > 0;
  wr5 : NOT (acyclic_mapped_item_usage(SELF));
  wr6 : SIZEOF(SELF.context_of_items.representations_in_context) = 1;
END_ENTITY;
```

```
ENTITY draughting_text_literal_with_delineation
 SUBTYPE OF (text_literal_with_delineation);
 WHERE
  wr1 : SELF.delineation IN ['underline','overline'];
END_ENTITY;


ENTITY draughting_title;
 items : SET[1:?] OF draughting_titled_item;
 language : label;
 contents : text;
END_ENTITY;


ENTITY drawing_definition;
 drawing_number : identifier;
 drawing_type : OPTIONAL label;
END_ENTITY;


ENTITY drawing_revision
 SUBTYPE OF (presentation_set);
 revision_identifier : identifier;
 drawing_identifier : drawing_definition;
 intended_scale : OPTIONAL text;
 UNIQUE
  ur1 : revision_identifier,drawing_identifier;
END_ENTITY;


ENTITY drawing_revision_sequence;
 predecessor : drawing_revision;
 successor : drawing_revision;
 WHERE
  wr1 : predecessor :<>: successor;
END_ENTITY;


ENTITY drawing_sheet_revision
 SUBTYPE OF (presentation_area);
 revision_identifier : identifier;
 WHERE
  wr1 : SIZEOF(QUERY(item <* SELF\representation.items | (
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN (TYPEOF(item))) AND (
        'AUTOMOTIVE_DESIGN.DRAWING_SHEET_REVISION' IN (TYPEOF(item\mapped_item
        .mapping_source.mapped_representation))))) = 0;
END_ENTITY;


ENTITY drawing_sheet_revision_usage
 SUBTYPE OF (area_in_set);
 sheet_number : identifier;
 UNIQUE
  ur1 : sheet_number,in_set;
```

```
 WHERE
  wr1 : ('AUTOMOTIVE_DESIGN.DRAWING_SHEET_REVISION' IN TYPEOF(SELF\area_in_set
        .area)) AND ('AUTOMOTIVE_DESIGN.DRAWING_REVISION' IN TYPEOF(SELF\
        area_in_set.in_set));
END_ENTITY;


ENTITY edge
 SUPERTYPE OF (ONEOF(edge_curve,oriented_edge))
 SUBTYPE OF (topological_representation_item);
 edge_start : vertex;
 edge_end : vertex;
END_ENTITY;


ENTITY edge_based_wireframe_model
 SUBTYPE OF (geometric_representation_item);
 ebwm_boundary : SET[1:?] OF connected_edge_set;
END_ENTITY;


ENTITY edge_based_wireframe_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(it <* SELF.items | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_MODEL',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM','AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D'
        ] * TYPEOF(it)) = 1))) = 0;
  wr2 : SIZEOF(QUERY(it <* SELF.items | SIZEOF([
        'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_MODEL',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM'] * TYPEOF(it)) = 1)) >= 1;
  wr3 : SIZEOF(QUERY(ebwm <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_MODEL' IN TYPEOF(it))) | NOT (
        SIZEOF(QUERY(eb <* ebwm\edge_based_wireframe_model.ebwm_boundary | NOT
        (SIZEOF(QUERY(edges <* eb.ces_edges | NOT (
        'AUTOMOTIVE_DESIGN.EDGE_CURVE' IN TYPEOF(edges)))) = 0))) = 0))) = 0;
  wr4 : SIZEOF(QUERY(ebwm <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_MODEL' IN TYPEOF(it))) | NOT (
        SIZEOF(QUERY(eb <* ebwm\edge_based_wireframe_model.ebwm_boundary | NOT
        (SIZEOF(QUERY(pline_edges <* QUERY(edges <* eb.ces_edges | (
        'AUTOMOTIVE_DESIGN.POLYLINE' IN TYPEOF(edges\edge_curve.edge_geometry)
        )) | NOT (SIZEOF(pline_edges\edge_curve.edge_geometry\polyline.points)
         > 2))) = 0))) = 0))) = 0;
  wr5 : SIZEOF(QUERY(ebwm <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_MODEL' IN TYPEOF(it))) | NOT (
        SIZEOF(QUERY(eb <* ebwm\edge_based_wireframe_model.ebwm_boundary | NOT
        (SIZEOF(QUERY(edges <* eb.ces_edges | NOT ((
        'AUTOMOTIVE_DESIGN.VERTEX_POINT' IN TYPEOF(edges.edge_start)) AND (
        'AUTOMOTIVE_DESIGN.VERTEX_POINT' IN TYPEOF(edges.edge_end))))) = 0)))
        = 0))) = 0;
  wr6 : SIZEOF(QUERY(ebwm <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_MODEL' IN TYPEOF(it))) | NOT (
        SIZEOF(QUERY(eb <* ebwm\edge_based_wireframe_model.ebwm_boundary | NOT
```

```
        (SIZEOF(QUERY(edges <* eb.ces_edges | NOT (valid_wireframe_edge_curve(
        edges\edge_curve.edge_geometry)))) = 0))) = 0))) = 0;
  wr7 : SIZEOF(QUERY(ebwm <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_MODEL' IN TYPEOF(it))) | NOT (
        SIZEOF(QUERY(eb <* ebwm\edge_based_wireframe_model.ebwm_boundary | NOT
        (SIZEOF(QUERY(edges <* eb.ces_edges | NOT ((
        valid_wireframe_vertex_point(edges.edge_start\vertex_point.
        vertex_geometry)) AND (valid_wireframe_vertex_point(edges.edge_end\
        vertex_point.vertex_geometry))))) = 0))) = 0))) = 0;
  wr8 : SIZEOF(QUERY(ebwm <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_MODEL' IN TYPEOF(it))) | NOT (
        SIZEOF(QUERY(eb <* ebwm\edge_based_wireframe_model.ebwm_boundary | NOT
        (SIZEOF(QUERY(con_edges <* QUERY(edges <* eb.ces_edges | (
        'AUTOMOTIVE_DESIGN.CONIC' IN TYPEOF(edges\edge_curve.edge_geometry)))
         | NOT ('AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D' IN TYPEOF(con_edges\
        edge_curve.edge_geometry\conic.position)))) = 0))) = 0))) = 0;
  wr9 : SIZEOF(QUERY(mi <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(it))) | NOT (
        'AUTOMOTIVE_DESIGN.' + 'EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION' IN
        TYPEOF(mi\mapped_item.mapping_source.mapped_representation)))) = 0;
  wr10 : SELF.context_of_items\geometric_representation_context.
         coordinate_space_dimension = 3;
END_ENTITY;


ENTITY edge_curve
 SUBTYPE OF (edge,geometric_representation_item);
 edge_geometry : curve;
 same_sense : BOOLEAN;
END_ENTITY;


ENTITY edge_loop
 SUBTYPE OF (loop,path);
 DERIVE
  ne : INTEGER := SIZEOF(SELF\path.edge_list);
 WHERE
  wr1 : (SELF\path.edge_list[1].edge_start) :=: (SELF\path.edge_list[ne].
        edge_end);
END_ENTITY;


ENTITY edge_round
 SUBTYPE OF (transition_feature);
 WHERE
  wr1 : (SELF.description <> 'constant radius') XOR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) > 0);
  wr2 : (SELF.description <> 'constant radius') OR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
```

```
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND ({1<=SIZEOF(pdr.used_representation.items)
        <=3}))) = 1);
  wr3 : (SELF.description <> 'constant radius') OR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | srwp_i.name IN ['radius',
        'first face offset','second face offset'])) = SIZEOF(pdr.
        used_representation.items)))) = 1);
  wr4 : (SELF.description <> 'constant radius') OR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'radius'))) = 1))) = 1);
  wr5 : (SELF.description <> 'constant radius') OR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'first face offset'))) <= 1))) = 1);
  wr6 : (SELF.description <> 'constant radius') OR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'second face offset'))) <= 1))) = 1);
END_ENTITY;


ENTITY effective_context_assignment
 ABSTRACT SUPERTYPE
 SUBTYPE OF (effectivity_assignment);
END_ENTITY;


ENTITY effective_item_assignment
 ABSTRACT SUPERTYPE
 SUBTYPE OF (effectivity_assignment);
END_ENTITY;


ENTITY effectivity
```

```
 SUPERTYPE OF (ONEOF(serial_numbered_effectivity,dated_effectivity,
               lot_effectivity));
 id : identifier;
 name : label;
 description : text;
END_ENTITY;


ENTITY effectivity_assignment
 ABSTRACT SUPERTYPE OF (ONEOF(effective_context_assignment,
                        effective_item_assignment));
 assigned_effectivity : effectivity;
 role : effectivity_role;
END_ENTITY;


ENTITY effectivity_relationship;
 name : label;
 description : text;
 relating_effectivity : effectivity;
 related_effectivity : effectivity;
END_ENTITY;


ENTITY effectivity_role;
 name : label;
 description : text;
END_ENTITY;


ENTITY electric_current_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.ELECTRIC_CURRENT_UNIT' IN TYPEOF(SELF\
        measure_with_unit.unit_component);
END_ENTITY;


ENTITY electric_current_unit
 SUBTYPE OF (named_unit);
 WHERE
  wr1 : (SELF\named_unit.dimensions.length_exponent = 0.0) AND (SELF\
        named_unit.dimensions.mass_exponent = 0.0) AND (SELF\named_unit.
        dimensions.time_exponent = 0.0) AND (SELF\named_unit.dimensions.
        electric_current_exponent = 1.0) AND (SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 0.0) AND (SELF\named_unit.
        dimensions.amount_of_substance_exponent = 0.0) AND (SELF\named_unit.
        dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;


ENTITY element_delivery
 SUBTYPE OF (directed_action);
 WHERE
```

```
  wr1 : SIZEOF(QUERY(aa <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
        'AUTOMOTIVE_DESIGN.ELEMENT_DELIVERY_ASSIGNMENT' IN TYPEOF(aa))) = 1;
  wr2 : (SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')) = 1) AND (SIZEOF(QUERY(aoa <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | aoa.role.name <>
        'destination')) = 0);
  wr3 : (SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY.DEFINITION')) <= 1) AND ((SIZEOF(USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'ACTION_PROPERTY.DEFINITION')) = 0) OR (SIZEOF(
        QUERY(ap <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY.DEFINITION') | SIZEOF(USEDIN(ap,'AUTOMOTIVE_DESIGN.'
        + 'ACTION_PROPERTY_REPRESENTATION.PROPERTY')) <> 1)) = 0) AND (SIZEOF(
        QUERY(ap <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY.DEFINITION') | SIZEOF(QUERY(apr <* USEDIN(ap,
        'AUTOMOTIVE_DESIGN.' + 'ACTION_PROPERTY_REPRESENTATION.PROPERTY') | (
        apr.representation.description <> 'delivery quantity') OR (SIZEOF(apr.
        items) <> 1) OR (SIZEOF(QUERY(apr_i <* apr.items | NOT (
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(apr_i)) OR (
        apr_i\measure_representation_item.name <> 'quantity'))) > 0))) > 0)) =
         0));
END_ENTITY;


ENTITY element_delivery_assignment
 SUBTYPE OF (action_assignment);
 items : SET[1:?] OF element_delivery_item;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.ELEMENT_DELIVERY' IN TYPEOF(SELF.assigned_action);
  wr2 : SELF.role.description = 'deliverable element';
  wr3 : (SIZEOF(SELF.items) = 1) AND (SIZEOF(QUERY(i <* SELF.items | NOT (
        'AUTOMOTIVE_DESIGN.APPLIED_ACTION_ASSIGNMENT' IN TYPEOF(i)) OR (i\
        applied_action_assignment.role.description <> 'activity element') OR (
        SIZEOF(i\applied_action_assignment.items) <> 1) OR (SIZEOF(QUERY(i_i <*
        i\applied_action_assignment.items |
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_FORMATION' IN TYPEOF(i_i))) +
        SIZEOF(QUERY(i_i <* i\applied_action_assignment.items | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i_i)) AND (i_i.name =
         'physical instance') AND (i_i.frame_of_reference.name =
        'physical occurrence'))) + SIZEOF(QUERY(i_i <* i\
        applied_action_assignment.items | (
        'AUTOMOTIVE_DESIGN.CONFIGURABLE_ITEM' IN TYPEOF(i_i)) AND (i_i.
        description = 'product specification'))) <> 1))) = 0);
END_ENTITY;


ENTITY elementary_surface
 SUPERTYPE OF (ONEOF(plane,cylindrical_surface,conical_surface,
               spherical_surface,toroidal_surface))
 SUBTYPE OF (surface);
 position : axis2_placement_3d;
END_ENTITY;
```

```
ENTITY ellipse
 SUBTYPE OF (conic);
 semi_axis_1 : positive_length_measure;
 semi_axis_2 : positive_length_measure;
END_ENTITY;

ENTITY evaluated_degenerate_pcurve
 SUBTYPE OF (degenerate_pcurve);
 equivalent_point : cartesian_point;
END_ENTITY;

ENTITY event_assignment
 SUBTYPE OF (action_assignment);
 items : SET[1:?] OF event_item;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.EVENT_REFERENCE' IN TYPEOF(SELF.assigned_action);
  wr2 : (SIZEOF(QUERY(i <* SELF.items |
        'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY' IN TYPEOF(i))) = 0) OR ((
        SIZEOF(QUERY(i <* SELF.items |
        'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY' IN TYPEOF(i))) > 0) AND (
        SELF.role.name IN ['effectivity start','effectivity end']));
  wr3 : (SIZEOF(QUERY(i <* SELF.items | 'AUTOMOTIVE_DESIGN.PROJECT' IN TYPEOF(
        i))) = 0) OR ((SIZEOF(QUERY(i <* SELF.items |
        'AUTOMOTIVE_DESIGN.PROJECT' IN TYPEOF(i))) > 0) AND (SELF.role.name IN
         ['actual start','planned start','planned end']));
END_ENTITY;

ENTITY event_based_effectivity
 SUBTYPE OF (effectivity);
 WHERE
  wr1 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT.ITEMS')) > 0;
  wr2 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT.ITEMS')) <= 2;
  wr3 : SIZEOF(QUERY(ea <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT.ITEMS') | ea.role.name =
        'effectivity start')) <= 1;
  wr4 : SIZEOF(QUERY(ea <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT.ITEMS') | ea.role.name =
        'effectivity end')) <= 1;
END_ENTITY;

ENTITY event_context_assignment
 SUBTYPE OF (action_assignment);
 items : SET[1:?] OF event_context_item;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.EVENT_REFERENCE' IN TYPEOF(SELF.assigned_action);
  wr2 : SELF.role.name = 'event context';
  wr3 : SIZEOF(SELF.items) = 1;
END_ENTITY;
```

```
ENTITY event_reference
 SUBTYPE OF (action);
 WHERE
  wr1 : SIZEOF(QUERY(aa <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
        'AUTOMOTIVE_DESIGN.EVENT_CONTEXT_ASSIGNMENT' IN TYPEOF(aa))) <= 1;
  wr2 : (SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY.DEFINITION')) <= 1) AND ((SIZEOF(USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'ACTION_PROPERTY.DEFINITION')) = 0) OR (SIZEOF(
        QUERY(ap <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY.DEFINITION') | (SIZEOF(USEDIN(ap,'AUTOMOTIVE_DESIGN.'
         + 'ACTION_PROPERTY_REPRESENTATION.PROPERTY')) = 1) AND (SIZEOF(QUERY(
        apr <* USEDIN(ap,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY_REPRESENTATION.PROPERTY') | (apr.representation.name
        <> 'offset definition') OR (SIZEOF(apr.representation.items) <> 1) OR
        (SIZEOF(QUERY(apr_r_i <* apr.representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.TIME_MEASURE_WITH_UNIT'] * TYPEOF(apr_r_i)) <> 2)
        OR (apr_r_i.name <> 'offset value'))) > 0))) = 0))) = 1));
END_ENTITY;


ENTITY exclusive_product_concept_feature_category
 SUBTYPE OF (product_concept_feature_category);
END_ENTITY;


ENTITY executed_action
 SUBTYPE OF (action);
END_ENTITY;


ENTITY expression_conversion_based_unit
 SUBTYPE OF (named_unit);
 name : label;
 conversion_basis : expression_definition;
END_ENTITY;


ENTITY expression_definition;
 name : label;
 description : text;
 operator : label;
 first_operand : expression_operand;
 second_operand : expression_operand;
END_ENTITY;


ENTITY extension
 SUBTYPE OF (derived_shape_aspect);
 WHERE
  wr1 : SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
END_ENTITY;
```

```
ENTITY external_identification_assignment
 ABSTRACT SUPERTYPE
 SUBTYPE OF (identification_assignment);
 source : external_source;
END_ENTITY;


ENTITY external_source;
 source_id : source_item;
 description : text;
END_ENTITY;


ENTITY externally_defined_classification_attribute
 SUBTYPE OF (externally_defined_item,representation);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.KNOWN_SOURCE' IN TYPEOF(SELF.source);
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') | (pdr.
        definition.description <> 'category attribute') OR NOT (
        'AUTOMOTIVE_DESIGN.CHARACTERIZED_PRODUCT_CATEGORY' IN TYPEOF(pdr.
        definition\property_definition.definition)))) = 0;
  wr3 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION')) >= 1;
  wr4 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_1')) >= 1;
  wr5 : SIZEOF(QUERY(rr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_1') | (rr.name <>
        'classification attribute value') OR NOT (
        'AUTOMOTIVE_DESIGN.PROPERTY_VALUE_REPRESENTATION' IN TYPEOF(rr.rep_2))
        )) = 0;
END_ENTITY;


ENTITY externally_defined_classification_system
 SUBTYPE OF (externally_defined_item,group);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.KNOWN_SOURCE' IN TYPEOF(SELF.source);
END_ENTITY;


ENTITY externally_defined_curve_font
 SUBTYPE OF (externally_defined_item);
END_ENTITY;


ENTITY externally_defined_hatch_style
 SUBTYPE OF (externally_defined_item,geometric_representation_item);
END_ENTITY;


ENTITY externally_defined_item;
 item_id : source_item;
 source : external_source;
```

```
END_ENTITY;


ENTITY externally_defined_product_category
 SUBTYPE OF (characterized_product_category,externally_defined_item);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.KNOWN_SOURCE' IN TYPEOF(SELF.source);
END_ENTITY;


ENTITY externally_defined_representation
 SUBTYPE OF (representation);
 WHERE
  wr1 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'DIGITAL_DOCUMENT_REFERENCE.ITEMS')) = 1;
  wr2 : SIZEOF(SELF.items) = 1;
  wr3 : SIZEOF(QUERY(it <* items | SIZEOF([
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_2D',
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D'] * TYPEOF(it)) = 1)) = 1;
END_ENTITY;


ENTITY externally_defined_symbol
 SUBTYPE OF (externally_defined_item);
END_ENTITY;


ENTITY externally_defined_text_font
 SUBTYPE OF (externally_defined_item);
END_ENTITY;


ENTITY externally_defined_thread
 SUBTYPE OF (externally_defined_item,feature_definition);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.used_representation.
        items) = 2))) = 1)) = 1;
  wr2 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i))
         AND (srwp_i.name = 'qualifier'))) = 1))) = 1)) = 1;
  wr3 : (SIZEOF(QUERY(adr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DOCUMENT_REFERENCE.ITEMS') | adr.role.name = 'specification')
        ) > 0) OR ('AUTOMOTIVE_DESIGN.KNOWN_SOURCE' IN TYPEOF(SELF.source));
END_ENTITY;
```

```
ENTITY externally_defined_tile_style
 SUBTYPE OF (externally_defined_item,geometric_representation_item);
END_ENTITY;


ENTITY extruded_face_solid
 SUBTYPE OF (swept_face_solid);
 extruded_direction : direction;
 depth : positive_length_measure;
 WHERE
  wr1 : dot_product((SELF\swept_face_solid.swept_face.face_geometry\
        elementary_surface.position.p[3]),extruded_direction) <> 0.0;
END_ENTITY;


ENTITY face
 SUPERTYPE OF (ONEOF(face_surface,oriented_face))
 SUBTYPE OF (topological_representation_item);
 bounds : SET[1:?] OF face_bound;
 WHERE
  wr1 : NOT (mixed_loop_type_set(list_to_set(list_face_loops(SELF))));
  wr2 : SIZEOF(QUERY(temp <* bounds | 'AUTOMOTIVE_DESIGN.FACE_OUTER_BOUND' IN
        TYPEOF(temp))) <= 1;
END_ENTITY;


ENTITY face_based_surface_model
 SUBTYPE OF (geometric_representation_item);
 fbsm_faces : SET[1:?] OF connected_face_set;
END_ENTITY;


ENTITY face_bound
 SUBTYPE OF (topological_representation_item);
 bound : loop;
 orientation : BOOLEAN;
END_ENTITY;


ENTITY face_outer_bound
 SUBTYPE OF (face_bound);
END_ENTITY;


ENTITY face_shape_representation
 SUBTYPE OF (representation);
 WHERE
  wr1 : SIZEOF(SELF.items) = 1;
  wr2 : SIZEOF(QUERY(it <* SELF.items | 'AUTOMOTIVE_DESIGN.FACE_SURFACE' IN
        TYPEOF(it))) = 1;
END_ENTITY;


ENTITY face_surface
 SUBTYPE OF (face,geometric_representation_item);
```

```
 face_geometry : surface;
 same_sense : BOOLEAN;
END_ENTITY;


ENTITY faceted_brep
 SUBTYPE OF (manifold_solid_brep);
END_ENTITY;


ENTITY faceted_brep_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(it <* items | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.FACETED_BREP','AUTOMOTIVE_DESIGN.MAPPED_ITEM',
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D'] * TYPEOF(it)) = 1))) = 0;
  wr2 : SIZEOF(QUERY(it <* items | SIZEOF(['AUTOMOTIVE_DESIGN.FACETED_BREP',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM'] * TYPEOF(it)) = 1)) > 0;
  wr3 : SIZEOF(QUERY(fbrep <* QUERY(it <* items |
        'AUTOMOTIVE_DESIGN.FACETED_BREP' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(
        csh <* msb_shells(fbrep) | NOT (SIZEOF(QUERY(fcs <* csh\
        connected_face_set.cfs_faces | NOT (('AUTOMOTIVE_DESIGN.FACE_SURFACE'
        IN TYPEOF(fcs)) AND (('AUTOMOTIVE_DESIGN.PLANE' IN TYPEOF(fcs\
        face_surface.face_geometry)) AND ('AUTOMOTIVE_DESIGN.CARTESIAN_POINT'
        IN TYPEOF(fcs\face_surface.face_geometry\elementary_surface.position.
        location)))))) = 0))) = 0))) = 0;
  wr4 : SIZEOF(QUERY(fbrep <* QUERY(it <* items |
        'AUTOMOTIVE_DESIGN.FACETED_BREP' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(
        csh <* msb_shells(fbrep) | NOT (SIZEOF(QUERY(fcs <* csh\
        connected_face_set.cfs_faces | NOT (SIZEOF(QUERY(bnds <* fcs.bounds |
        'AUTOMOTIVE_DESIGN.FACE_OUTER_BOUND' IN TYPEOF(bnds))) = 1))) = 0))) =
         0))) = 0;
  wr5 : SIZEOF(QUERY(msb <* QUERY(it <* items |
        'AUTOMOTIVE_DESIGN.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) |
        'AUTOMOTIVE_DESIGN.ORIENTED_CLOSED_SHELL' IN TYPEOF(msb\
        manifold_solid_brep.outer))) = 0;
  wr6 : SIZEOF(QUERY(brv <* QUERY(it <* items |
        'AUTOMOTIVE_DESIGN.BREP_WITH_VOIDS' IN TYPEOF(it)) | NOT (SIZEOF(QUERY
        (csh <* brv\brep_with_voids.voids | csh\oriented_closed_shell.
        orientation)) = 0))) = 0;
  wr7 : SIZEOF(QUERY(mi <* QUERY(it <* items | 'AUTOMOTIVE_DESIGN.MAPPED_ITEM'
         IN TYPEOF(it)) | NOT (
        'AUTOMOTIVE_DESIGN.FACETED_BREP_SHAPE_REPRESENTATION' IN TYPEOF(mi\
        mapped_item.mapping_source.mapped_representation)))) = 0;
END_ENTITY;


ENTITY feature_component_definition
 SUBTYPE OF (characterized_object);
 WHERE
  wr1 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(sa <* USEDIN(pds,
        'AUTOMOTIVE_DESIGN.' + 'SHAPE_ASPECT.OF_SHAPE') | NOT (
```

```
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(sa)))) > 1)) =
        0;
END_ENTITY;


ENTITY feature_component_relationship
 SUBTYPE OF (shape_aspect_relationship);
END_ENTITY;


ENTITY feature_definition
 SUBTYPE OF (characterized_object);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | 'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF
        (pdr.used_representation))) = 1)) = 1;
  wr2 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(
        srwp_i)) AND (srwp_i.name = 'orientation'))) = 1))) = 1)) = 1;
END_ENTITY;


ENTITY feature_in_panel
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SELF.description IN ['pocket','boss','stairstep'];
  wr2 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.used_representation.
        items) = 3))) = 1)) = 1;
  wr3 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'base fillet radius'))) = 1))) = 1)) = 1;
  wr4 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
```

```
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'target fillet radius'))) = 1))) = 1)) = 1;
  wr5 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMTOIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'base boundary') AND (sa.
        description IN ['directed angle occurrence',
        'closed profile occurrence']))) = 1))) = 1;
  wr6 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMTOIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'target boundary') AND (sa.
        description IN ['directed angle occurrence',
        'closed profile occurrence']))) = 1))) = 1;
  wr7 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMTOIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | sa.description =
        'directed angle occurrence')) <= 1))) <= 1;
  wr8 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | NOT (
        'AUTOMTOIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) OR (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.description =
        'directed angle occurrence') AND (SIZEOF(QUERY(sdr <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sdr)) AND (
        sdr.name = 'directed angle usage') AND (
        'AUTOMOTIVE_DESIGN.DIRECTED_ANGLE' IN TYPEOF(sdr.relating_shape_aspect
        )))) <> 1))) = 0))) = 0;
  wr9 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | NOT (
        'AUTOMTOIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) OR (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.description =
        'closed profile occurrence') AND (SIZEOF(QUERY(sdr <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sdr)) AND (
        sdr.name = 'closed profile usage') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.NGON_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.CLOSED_PATH_PROFILE'] * TYPEOF(sdr.
        relating_shape_aspect)) = 1))) <> 1))) = 0))) = 0;
  wr10 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY(pdr <* USEDIN(pd,
```

```
         'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
         'AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr
         .used_representation)))) = 0) OR (SIZEOF(QUERY(pdr <* USEDIN(pd,
         'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
         'AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr
         .used_representation)) AND (pdr.used_representation.description =
         'target face'))) = 1))) <= 1;
  wr11 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION.DEFINITION') | (
         'AUTOMTOIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
         SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
         'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'target boundary') AND (sa.
         description = 'directed angle occurrence') AND (SIZEOF(QUERY(pdr <*
         USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         'AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr
         .used_representation))) = 1))) = 1))) = 1;
END_ENTITY;


ENTITY feature_pattern
 SUBTYPE OF (replicate_feature);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(
        srwp_i)) AND (srwp_i.name = 'added instance placement'))) > 0))) = 1))
         = 1;
  wr2 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | NOT ('AUTOMOTIVE_DESIGN.PLACEMENT' IN
        TYPEOF(srwp_i)))) > 0))) > 0)) = 0;
END_ENTITY;


ENTITY featured_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(rr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'REPRESENTATION_RELATIONSHIP.REP_2') | (rr.name =
        'feature application sequence') AND (SIZEOF(QUERY(pdr <* USEDIN(rr.
        rep_1,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pdr.definition)
        )) = 1))) = 1;
```

```
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pdr.definition)
        )) = 1;
END_ENTITY;


ENTITY fill_area_style;
 name : label;
 fill_styles : SET[1:?] OF fill_style_select;
 WHERE
  wr1 : SIZEOF(QUERY(fill_style <* SELF.fill_styles | 'AUTOMOTIVE_DESIGN.' +
        'FILL_AREA_STYLE_COLOUR' IN TYPEOF(fill_style))) <= 1;
END_ENTITY;


ENTITY fill_area_style_colour;
 name : label;
 fill_colour : colour;
END_ENTITY;


ENTITY fill_area_style_hatching
 SUBTYPE OF (geometric_representation_item);
 hatch_line_appearance : curve_style;
 start_of_next_hatch_line : one_direction_repeat_factor;
 point_of_reference_hatch_line : cartesian_point;
 pattern_start : cartesian_point;
 hatch_line_angle : plane_angle_measure;
END_ENTITY;


ENTITY fill_area_style_tile_symbol_with_style
 SUBTYPE OF (geometric_representation_item);
 symbol : annotation_symbol_occurrence;
END_ENTITY;


ENTITY fill_area_style_tiles
 SUBTYPE OF (geometric_representation_item);
 tiling_pattern : two_direction_repeat_factor;
 tiles : SET[1:?] OF fill_area_style_tile_shape_select;
 tiling_scale : positive_ratio_measure;
END_ENTITY;


ENTITY fillet
 SUBTYPE OF (transition_feature);
 WHERE
  wr1 : (SELF.description <> 'constant radius') XOR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) > 0);
```

```
  wr2 : (SELF.description <> 'constant radius') OR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND ({1<=SIZEOF(pdr.used_representation.items)
        <=3}))) = 1);
  wr3 : (SELF.description <> 'constant radius') OR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | srwp_i.name IN ['radius',
        'first face offset','second face offset'])) = SIZEOF(pdr.
        used_representation.items)))) = 1);
  wr4 : (SELF.description <> 'constant radius') OR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'radius'))) = 1))) = 1);
  wr5 : (SELF.description <> 'constant radius') OR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'first face offset'))) <= 1))) = 1);
  wr6 : (SELF.description <> 'constant radius') OR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'second face offset'))) <= 1))) = 1);
END_ENTITY;


ENTITY flatness_tolerance
 SUBTYPE OF (geometric_tolerance);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE'
         IN TYPEOF(SELF));
END_ENTITY;


ENTITY fully_constrained_pair
```

```
 SUBTYPE OF (kinematic_pair);
END_ENTITY;


ENTITY functionally_defined_transformation;
 name : label;
 description : text;
END_ENTITY;


ENTITY gear_pair
 SUBTYPE OF (kinematic_pair);
 radius_first_link : length_measure;
 radius_second_link : length_measure;
 bevel : plane_angle_measure;
 helical_angle : plane_angle_measure;
 gear_ratio : REAL;
END_ENTITY;


ENTITY gear_pair_range
 SUBTYPE OF (simple_pair_range);
 SELF\simple_pair_range.applies_to_pair : gear_pair;
 lower_limit_actual_rotation_1 : rotational_range_measure;
 upper_limit_actual_rotation_1 : rotational_range_measure;
 WHERE
  wr1 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        lower_limit_actual_rotation_1)) OR (
        'AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        upper_limit_actual_rotation_1))) XOR (lower_limit_actual_rotation_1 <
        upper_limit_actual_rotation_1);
END_ENTITY;


ENTITY gear_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : gear_pair;
 actual_rotation_1 : plane_angle_measure;
 DERIVE
  actual_rotation_2 : plane_angle_measure := - actual_rotation_1 * SELF\
                      pair_value.applies_to_pair\gear_pair.gear_ratio;
END_ENTITY;


ENTITY geometric_alignment
 SUBTYPE OF (derived_shape_aspect);
 WHERE
  wr1 : SIZEOF(SELF\derived_shape_aspect.deriving_relationships) > 1;
END_ENTITY;


ENTITY geometric_curve_set
 SUBTYPE OF (geometric_set);
 WHERE
```

```
  wr1 : SIZEOF(QUERY(temp <* SELF\geometric_set.elements |
        'AUTOMOTIVE_DESIGN.SURFACE' IN TYPEOF(temp))) = 0;
END_ENTITY;


ENTITY geometric_intersection
 SUBTYPE OF (derived_shape_aspect);
 WHERE
  wr1 : SIZEOF(SELF\derived_shape_aspect.deriving_relationships) > 1;
END_ENTITY;


ENTITY geometric_representation_context
 SUBTYPE OF (representation_context);
 coordinate_space_dimension : dimension_count;
END_ENTITY;


ENTITY geometric_representation_item
 SUPERTYPE OF (ONEOF(point,direction,vector,placement,
               cartesian_transformation_operator,curve,surface,edge_curve,
               face_surface,poly_loop,vertex_point,solid_model,boolean_result,
               sphere,right_circular_cone,right_circular_cylinder,torus,block,
               right_angular_wedge,half_space_solid,shell_based_surface_model,
               face_based_surface_model,edge_based_wireframe_model,
               geometric_set))
 SUBTYPE OF (representation_item);
 DERIVE
  dim : dimension_count := dimension_of(SELF);
 WHERE
  wr1 : SIZEOF(QUERY(using_rep <* using_representations(SELF) | NOT (
        'AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_CONTEXT' IN TYPEOF(
        using_rep.context_of_items)))) = 0;
END_ENTITY;


ENTITY geometric_set
 SUPERTYPE OF (geometric_curve_set)
 SUBTYPE OF (geometric_representation_item);
 elements : SET[1:?] OF geometric_set_select;
END_ENTITY;


ENTITY geometric_tolerance;
 name : label;
 description : text;
 magnitude : measure_with_unit;
 toleranced_shape_aspect : shape_aspect;
 WHERE
  wr1 : magnitude.value_component >= 0.0;
END_ENTITY;


ENTITY geometric_tolerance_relationship;
```

```
 name : label;
 description : text;
 relating_geometric_tolerance : geometric_tolerance;
 related_geometric_tolerance : geometric_tolerance;
END_ENTITY;


ENTITY geometric_tolerance_with_datum_reference
 SUBTYPE OF (geometric_tolerance);
 datum_system : SET[1:?] OF datum_reference;
END_ENTITY;


ENTITY geometric_tolerance_with_defined_unit
 SUBTYPE OF (geometric_tolerance);
 unit_size : measure_with_unit;
 WHERE
  wr1 : unit_size.value_component > 0.0;
END_ENTITY;


ENTITY geometrical_tolerance_callout
 SUBTYPE OF (draughting_callout);
END_ENTITY;


ENTITY geometrically_bounded_2d_wireframe_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SELF.context_of_items\geometric_representation_context.
        coordinate_space_dimension = 2;
  wr2 : SIZEOF(QUERY(item <* SELF.items | NOT (SIZEOF(TYPEOF(item) * [
        'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET',
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_2D','AUTOMOTIVE_DESIGN.MAPPED_ITEM'
        ]) = 1))) = 0;
  wr3 : SIZEOF(QUERY(item <* SELF.items | SIZEOF(TYPEOF(item) * [
        'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM']) = 1)) >= 1;
  wr4 : SIZEOF(QUERY(mi <* QUERY(item <* SELF.items | (
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(item))) | NOT (
        'AUTOMOTIVE_DESIGN.' +
        'GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION' IN TYPEOF(mi\
        mapped_item.mapping_source.mapped_representation)))) = 0;
  wr5 : SIZEOF(QUERY(gcs <* QUERY(item <* SELF.items | (
        'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET' IN TYPEOF(item))) | NOT (
        SIZEOF(QUERY(elem <* gcs\geometric_set.elements | NOT (SIZEOF(TYPEOF(
        elem) * ['AUTOMOTIVE_DESIGN.B_SPLINE_CURVE','AUTOMOTIVE_DESIGN.CIRCLE'
        ,'AUTOMOTIVE_DESIGN.COMPOSITE_CURVE','AUTOMOTIVE_DESIGN.ELLIPSE',
        'AUTOMOTIVE_DESIGN.OFFSET_CURVE_2D','AUTOMOTIVE_DESIGN.POINT',
        'AUTOMOTIVE_DESIGN.POLYLINE','AUTOMOTIVE_DESIGN.TRIMMED_CURVE']) = 1))
        ) = 0))) = 0;
  wr6 : SIZEOF(QUERY(gcs <* QUERY(item <* SELF.items | (
        'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET' IN TYPEOF(item))) | NOT (
        SIZEOF(QUERY(crv <* QUERY(elem <* gcs\geometric_set.elements | (
```

```
        'AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF(elem))) | NOT (
        valid_basis_curve_in_2d_wireframe(crv,
        'AIC_GEOMETRICALLY_BOUNDED_2D_WIREFRAME')))) = 0))) = 0;
  wr7 : SIZEOF(QUERY(gcs <* QUERY(item <* SELF.items | (
        'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET' IN TYPEOF(item))) | NOT (
        SIZEOF(QUERY(pnt <* QUERY(elem <* gcs\geometric_set.elements | (
        'AUTOMOTIVE_DESIGN.POINT' IN TYPEOF(elem))) | NOT (SIZEOF(TYPEOF(pnt)
        * ['AUTOMOTIVE_DESIGN.CARTESIAN_POINT',
        'AUTOMOTIVE_DESIGN.POINT_ON_CURVE']) = 1))) = 0))) = 0;
  wr8 : SIZEOF(QUERY(gcs <* QUERY(item <* SELF.items | (
        'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET' IN TYPEOF(item))) | NOT (
        SIZEOF(QUERY(pl <* QUERY(elem <* gcs\geometric_set.elements | (
        'AUTOMOTIVE_DESIGN.POLYLINE' IN TYPEOF(elem))) | NOT (SIZEOF(pl\
        polyline.points) > 2))) = 0))) = 0;
END_ENTITY;


ENTITY geometrically_bounded_surface_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(it <* SELF.items | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.GEOMETRIC_SET','AUTOMOTIVE_DESIGN.MAPPED_ITEM',
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D'] * TYPEOF(it)) = 1))) = 0;
  wr2 : SIZEOF(QUERY(it <* SELF.items | SIZEOF([
        'AUTOMOTIVE_DESIGN.GEOMETRIC_SET','AUTOMOTIVE_DESIGN.MAPPED_ITEM'] *
        TYPEOF(it)) = 1)) > 0;
  wr3 : SIZEOF(QUERY(mi <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(it)) | NOT (
        'AUTOMOTIVE_DESIGN.' +
        'GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION' IN TYPEOF(mi\
        mapped_item.mapping_source.mapped_representation)))) = 0;
  wr4 : SIZEOF(QUERY(gs <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.GEOMETRIC_SET' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(
        pnt <* QUERY(gsel <* gs\geometric_set.elements |
        'AUTOMOTIVE_DESIGN.POINT' IN TYPEOF(gsel)) | NOT (gbsf_check_point(pnt
        )))) = 0))) = 0;
  wr5 : SIZEOF(QUERY(gs <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.GEOMETRIC_SET' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(
        cv <* QUERY(gsel <* gs\geometric_set.elements |
        'AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF(gsel)) | NOT (gbsf_check_curve(cv)
        ))) = 0))) = 0;
  wr6 : SIZEOF(QUERY(gs <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.GEOMETRIC_SET' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(
        sf <* QUERY(gsel <* gs\geometric_set.elements |
        'AUTOMOTIVE_DESIGN.SURFACE' IN TYPEOF(gsel)) | NOT (gbsf_check_surface
        (sf)))) = 0))) = 0;
END_ENTITY;


ENTITY geometrically_bounded_wireframe_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(it <* SELF.items | NOT (SIZEOF(TYPEOF(it) * [
```

```
        'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET',
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D','AUTOMOTIVE_DESIGN.MAPPED_ITEM'
        ]) = 1))) = 0;
  wr2 : SIZEOF(QUERY(it <* SELF.items | SIZEOF(TYPEOF(it) * [
        'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM']) = 1)) >= 1;
  wr3 : SIZEOF(QUERY(gcs <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET' IN TYPEOF(it))) | NOT (SIZEOF(
        QUERY(crv <* QUERY(elem <* gcs\geometric_set.elements | (
        'AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF(elem))) | NOT (
        valid_geometrically_bounded_wf_curve(crv)))) = 0))) = 0;
  wr4 : SIZEOF(QUERY(gcs <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET' IN TYPEOF(it))) | NOT (SIZEOF(
        QUERY(pnts <* QUERY(elem <* gcs\geometric_set.elements | (
        'AUTOMOTIVE_DESIGN.POINT' IN TYPEOF(elem))) | NOT (
        valid_geometrically_bounded_wf_point(pnts)))) = 0))) = 0;
  wr5 : SIZEOF(QUERY(gcs <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET' IN TYPEOF(it))) | NOT (SIZEOF(
        QUERY(cnc <* QUERY(elem <* gcs\geometric_set.elements | (
        'AUTOMOTIVE_DESIGN.CONIC' IN TYPEOF(elem))) | NOT (
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D' IN TYPEOF(cnc\conic.position)))
        ) = 0))) = 0;
  wr6 : SIZEOF(QUERY(gcs <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.GEOMETRIC_CURVE_SET' IN TYPEOF(it))) | NOT (SIZEOF(
        QUERY(pline <* QUERY(elem <* gcs\geometric_set.elements | (
        'AUTOMOTIVE_DESIGN.POLYLINE' IN TYPEOF(elem))) | NOT (SIZEOF(pline\
        polyline.points) > 2))) = 0))) = 0;
  wr7 : SIZEOF(QUERY(mi <* QUERY(it <* SELF.items | (
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(it))) | NOT (
        'AUTOMOTIVE_DESIGN.' +
        'GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION' IN TYPEOF(mi\
        mapped_item.mapping_source.mapped_representation)))) = 0;
  wr8 : SELF.context_of_items\geometric_representation_context.
        coordinate_space_dimension = 3;
END_ENTITY;


ENTITY global_uncertainty_assigned_context
 SUBTYPE OF (representation_context);
 uncertainty : SET[1:?] OF uncertainty_measure_with_unit;
END_ENTITY;


ENTITY global_unit_assigned_context
 SUBTYPE OF (representation_context);
 units : SET[1:?] OF unit;
END_ENTITY;


ENTITY group;
 name : label;
 description : text;
END_ENTITY;
```

```
ENTITY group_assignment
 ABSTRACT SUPERTYPE;
 assigned_group : group;
 role : group_role;
END_ENTITY;


ENTITY group_member_assignment
 SUBTYPE OF (group_assignment);
 items : SET[1:?] OF group_member_item;
 WHERE
  wr1 : (SIZEOF(QUERY(i <* SELF.items | 'AUTOMOTIVE_DESIGN.SHAPE_ASPECT' IN
        TYPEOF(i))) = 0) OR (SELF.role.name = 'feature categorization');
  wr2 : (SIZEOF(QUERY(i <* SELF.items | SIZEOF([
        'AUTOMOTIVE_DESIGN.DOCUMENT_TYPE',
        'AUTOMOTIVE_DESIGN.DRAWING_DEFINITION',
        'AUTOMOTIVE_DESIGN.APPROVAL_STATUS',
        'AUTOMOTIVE_DESIGN.SECURITY_CLASSIFICATION_LEVEL',
        'AUTOMOTIVE_DESIGN.CHARACTERIZED_PRODUCT_CATEGORY'] * TYPEOF(i)) = 1))
         = 0) OR ((SIZEOF(['AUTOMOTIVE_DESIGN.CLASSIFICATION_SYSTEM',
        'AUTOMOTIVE_DESIGN.EXTERNALLY_DEFINED_CLASSIFICATION_SYSTEM'] * TYPEOF
        (SELF.assigned_group)) = 1) AND (SELF.role.name = 'classified member')
        );
  wr3 : (SIZEOF(QUERY(i <* SELF.items | 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION'
         IN TYPEOF(i))) = 0) OR (SIZEOF(QUERY(i <* SELF.items | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)) AND ((i\
        product_definition.frame_of_reference.name <> 'definitional template')
         OR ((SIZEOF(QUERY(pdr <* USEDIN(i,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') | (
        'AUTOMOTIVE_DESIGN.ASSEMBLY_COMPONENT_RELATIONSHIP' IN TYPEOF(pdr))
        AND (pdr.name = 'mating membership'))) + SIZEOF(QUERY(pdr <* USEDIN(i,
        'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') | (pdr.
        name = 'collection membership')))) <> 1)))) = 0);
  wr4 : (SIZEOF(QUERY(i <* SELF.items | 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION'
         IN TYPEOF(i))) = 0) OR (SIZEOF(QUERY(i <* SELF.items | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)) AND (SIZEOF(QUERY
        (pdr <* USEDIN(i,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
        'AUTOMOTIVE_DESIGN.ASSEMBLY_COMPONENT_RELATIONSHIP' IN TYPEOF(pdr))) =
         0))) = SIZEOF(QUERY(i <* SELF.items |
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)))) OR ((SIZEOF(
        QUERY(i <* SELF.items | ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN
        TYPEOF(i)) AND (SIZEOF(QUERY(pdr <* USEDIN(i,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
        'AUTOMOTIVE_DESIGN.ASSEMBLY_COMPONENT_RELATIONSHIP' IN TYPEOF(pdr))) >
         0))) > 0) AND (SELF.assigned_group.name = 'assembly_definition') AND
        (SELF.role.name = 'assembly'));
  wr5 : (SIZEOF(QUERY(i <* SELF.items | 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION'
         IN TYPEOF(i))) = 0) OR (SIZEOF(QUERY(i <* SELF.items | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)) AND (SIZEOF(QUERY
        (pdr <* USEDIN(i,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') | pdr.
```

```
        name = 'mating membership')) = 0))) = SIZEOF(QUERY(i <* SELF.items |
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)))) OR ((SIZEOF(
        QUERY(i <* SELF.items | ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN
        TYPEOF(i)) AND (SIZEOF(QUERY(pdr <* USEDIN(i,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') | pdr.
        name = 'mating membership')) > 0))) > 0) AND (SELF.assigned_group.name
         = 'mating definition') AND (SELF.role.name = 'mating'));
  wr6 : (SIZEOF(QUERY(i <* SELF.items | 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION'
         IN TYPEOF(i))) = 0) OR (SIZEOF(QUERY(i <* SELF.items | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)) AND (SIZEOF(QUERY
        (pdr <* USEDIN(i,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') | pdr.
        name = 'collection membership')) = 0))) = SIZEOF(QUERY(i <* SELF.items
         | 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(i)))) OR ((SIZEOF(
        QUERY(i <* SELF.items | ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN
        TYPEOF(i)) AND (SIZEOF(QUERY(pdr <* USEDIN(i,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') | pdr.
        name = 'collection membership')) > 0))) > 0) AND (SELF.assigned_group.
        name = 'collection definition') AND (SELF.role.name = 'collection'));
  wr7 : (SIZEOF(QUERY(i <* SELF.items |
        'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_FEATURE_CATEGORY_USAGE' IN TYPEOF(i
        ))) = 0) OR ((SELF.assigned_group.name = 'influencing category group')
         AND (SELF.role.name = 'class usage group member'));
  wr8 : (SIZEOF(QUERY(i <* SELF.items |
        'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_FEATURE' IN TYPEOF(i))) = 0) OR ((
        'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_FEATURE_CATEGORY' IN TYPEOF(SELF.
        assigned_group)) AND (SELF.role.name = 'specification category member'
        ));
END_ENTITY;


ENTITY group_relationship;
 name : label;
 description : text;
 relating_group : group;
 related_group : group;
END_ENTITY;


ENTITY group_role;
 name : label;
 description : text;
END_ENTITY;


ENTITY half_space_solid
 SUBTYPE OF (geometric_representation_item);
 base_surface : surface;
 agreement_flag : BOOLEAN;
END_ENTITY;


ENTITY hidden_element_over_riding_styled_item
 SUBTYPE OF (context_dependent_over_riding_styled_item);
```

```
 WHERE
  wr1 : SIZEOF(SELF.style_context) = 1;
  wr2 : 'AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN TYPEOF(SELF.style_context[1])
        ;
  wr3 : 'AUTOMOTIVE_DESIGN.CAMERA_IMAGE' IN TYPEOF(SELF.item);
  wr4 : 'AUTOMOTIVE_DESIGN.CAMERA_MODEL_D3_WITH_HLHSR' IN TYPEOF(SELF.item\
        mapped_item.mapping_source.mapping_origin);
  wr5 : SIZEOF(QUERY(rep <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.REPRESENTATION.ITEMS') |
        'AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN TYPEOF(rep))) > 0;
END_ENTITY;


ENTITY hole_bottom
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SELF.description IN ['through','flat','flat with chamfer',
        'flat with radius','spherical','conical'];
  wr3 : NOT (SELF.description IN ['through','flat with chamfer']) OR (SIZEOF(
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 0);
  wr4 : NOT (SELF.description IN ['flat','flat with radius','spherical',
        'conical']) OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1);
  wr5 : (SELF.description <> 'flat') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.used_representation.
        items) = 1))) = 1);
  wr6 : NOT (SELF.description IN ['flat with radius','spherical']) OR (SIZEOF(
        QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(pdr.used_representation.items) = 2)
        )) = 1);
  wr7 : (SELF.description <> 'conical') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.used_representation.
        items) = 3))) = 1);
  wr8 : NOT (SELF.description IN ['flat','flat with radius','spherical',
        'conical']) OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'bottom location') AND (
        srwp_i.description IN ['part start','part end']) AND (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i))
        )) = 1))) = 1);
```

```
wr9 : (SELF.description <> 'flat with radius') OR (SIZEOF(QUERY(pdr <* USEDIN
      (SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
      .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (srwp_i.name = 'corner radius') AND (
      SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
       = 1))) = 1);
wr10 : (SELF.description <> 'spherical') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF
       ,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
       'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
       pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
       used_representation.items | (srwp_i.name = 'radius') AND (SIZEOF([
       'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2))
       ) = 1))) = 1);
wr11 : (SELF.description <> 'conical') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
       'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
       'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
       pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
       used_representation.items | (srwp_i.name = 'tip radius') AND (SIZEOF(
       ['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2))
       ) = 1))) = 1);
wr12 : (SELF.description <> 'conical') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
       'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
       'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
       pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
       used_representation.items | (srwp_i.name = 'tip angle') AND (SIZEOF([
       'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i))
       = 2))) = 1))) = 1);
wr13 : (SELF.description <> 'flat with chamfer') OR (SIZEOF(QUERY(fcr <*
       USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (
       'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr))
       AND (fcr.name = 'first chamfer offset') AND (
       'AUTOMOTIVE_DESIGN.CHAMFER_OFFSET' IN TYPEOF(fcr.related_shape_aspect
       )))) = 1);
wr14 : (SELF.description <> 'flat with chamfer') OR (SIZEOF(QUERY(fcr <*
       USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (
       'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr))
       AND (fcr.name = 'second chamfer offset') AND (
       'AUTOMOTIVE_DESIGN.CHAMFER_OFFSET' IN TYPEOF(fcr.related_shape_aspect
       )))) = 1);
wr15 : (SIZEOF(QUERY(fcr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (
```

```
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr))
        AND (fcr.description = 'hole bottom usage') AND (
        'AUTOMOTIVE_DESIGN.ROUND_HOLE' IN TYPEOF(fcr.related_shape_aspect.
        of_shape.definition)) AND (fcr.related_shape_aspect.of_shape.
        definition.description = 'bottom condition occurrence'))) >= 1);
END_ENTITY;


ENTITY hole_in_panel
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'hole profile') AND (sa.
        description = 'closed profile occurrence') AND (SIZEOF(QUERY(sdr <*
        USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sdr)) AND (
        sdr.name = 'closed profile usage') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.NGON_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.CLOSED_PATH_PROFILE'] * TYPEOF(sdr.
        relating_shape_aspect)) = 1))) = 1))) = 1))) = 1;
  wr2 : (SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'taper') AND (sa.description =
        'taper occurrence'))) = 0))) = 0) OR (SIZEOF(QUERY(pds <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'taper') AND (sa.description =
        'taper occurrence') AND (SIZEOF(QUERY(fcr <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr)) AND
         (fcr.name = 'taper usage') AND ('AUTOMOTIVE_DESIGN.TAPER' IN TYPEOF(
        fcr.relating_shape_aspect)))) = 1))) = 1))) = 1);
  wr3 : (SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'barring profile') AND (sa.
        description = 'closed profile occurrence'))) = 0))) = 0) OR (SIZEOF(
        QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'barring profile') AND (sa.
```

```
        description = 'closed profile occurrence') AND (SIZEOF(QUERY(sdr <*
        USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sdr)) AND (
        sdr.name = 'closed profile usage'))) = 1))) = 1))) = 1);
END_ENTITY;


ENTITY hybrid_3d_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(i <* SELF\representation.items | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM'] * TYPEOF(i)) = 1))) = 0;
  wr2 : SIZEOF(QUERY(mi <* QUERY(item <* SELF\representation.items |
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(item)) | NOT (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN TYPEOF(mi\mapped_item.
        mapping_source.mapped_representation)))) = 0;
END_ENTITY;


ENTITY hyperbola
 SUBTYPE OF (conic);
 semi_axis : positive_length_measure;
 semi_imag_axis : positive_length_measure;
END_ENTITY;


ENTITY identification_assignment
 ABSTRACT SUPERTYPE;
 assigned_id : identifier;
 role : identification_role;
END_ENTITY;


ENTITY identification_role;
 name : label;
 description : text;
END_ENTITY;


ENTITY initial_state;
 applies_to_mechanism : mechanism;
 pair_values : SET[1:?] OF pair_value;
 WHERE
  wr1 : SIZEOF(QUERY(joint <* applies_to_mechanism.structure_definition.joints
         | SIZEOF(QUERY(init_val <* pair_values | init_val.applies_to_pair.
        joint :=: joint)) <> 1)) = 0;
END_ENTITY;


ENTITY instanced_feature
 SUBTYPE OF (feature_definition,shape_aspect);
 WHERE
```

```
  wr1 : 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(SELF.of_shape.
        definition);
END_ENTITY;


ENTITY interpolated_configuration_sequence;
 interpolation : SET[1:?] OF configuration_interpolation;
END_ENTITY;


ENTITY intersection_curve
 SUBTYPE OF (surface_curve);
 WHERE
  wr1 : SIZEOF(SELF\surface_curve.associated_geometry) = 2;
  wr2 : associated_surface(SELF\surface_curve.associated_geometry[1]) <>
        associated_surface(SELF\surface_curve.associated_geometry[2]);
END_ENTITY;


ENTITY invisibility;
 invisible_items : SET[1:?] OF invisible_item;
END_ENTITY;


ENTITY item_defined_transformation
 SUBTYPE OF (representation_item_relationship);
 DERIVE
  transform_item_1 : representation_item := SELF\
                     representation_item_relationship.
                     relating_representation_item;
  transform_item_2 : representation_item := SELF\
                     representation_item_relationship.
                     related_representation_item;
END_ENTITY;


ENTITY joggle
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.used_representation.
        items) <= 6))) = 1)) = 1;
  wr2 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF(['AUTOMOTIVE_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM','AUTOMOTIVE_DESIGN.' +
        'PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2) AND (srwp_i.
        name = 'direction angle at path end'))) = 1))) = 1)) = 1;
```

```
wr3 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (SIZEOF(['AUTOMOTIVE_DESIGN.' +
      'MEASURE_REPRESENTATION_ITEM','AUTOMOTIVE_DESIGN.' +
      'PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2) AND (srwp_i.
      name = 'direction angle at path start'))) = 1))) = 1)) = 1;
wr4 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (SIZEOF(['AUTOMOTIVE_DESIGN.' +
      'MEASURE_REPRESENTATION_ITEM','AUTOMOTIVE_DESIGN.' +
      'LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2) AND (srwp_i.name =
      'height'))) = 1))) = 1)) = 1;
wr5 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (SIZEOF(['AUTOMOTIVE_DESIGN.' +
      'MEASURE_REPRESENTATION_ITEM','AUTOMOTIVE_DESIGN.' +
      'PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2) AND (srwp_i.
      name = 'slope angle at path end'))) = 1))) = 1)) = 1;
wr6 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (SIZEOF(['AUTOMOTIVE_DESIGN.' +
      'MEASURE_REPRESENTATION_ITEM','AUTOMOTIVE_DESIGN.' +
      'PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2) AND (srwp_i.
      name = 'slope angle at path start'))) = 1))) = 1)) = 1;
wr7 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
      SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'defining path') AND (sa.
      description = 'path feature component occurrence') AND (SIZEOF(QUERY(
      sdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
      'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sdr)) AND (
      sdr.name = 'path feature component usage') AND (
      'AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN TYPEOF(sdr.
      relating_shape_aspect)) AND (sdr.relating_shape_aspect.description =
      'linear'))) = 1))) = 1))) = 1;
wr8 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
```

```
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'termination') AND (sa.
        description = 'joggle termination occurrence') AND (SIZEOF(QUERY(sdr <*
        USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sdr)) AND (
        sdr.name = 'joggle termination usage') AND (
        'AUTOMOTIVE_DESIGN.JOGGLE_TERMINATION' IN TYPEOF(sdr.
        relating_shape_aspect)))) = 1))) = 1))) = 1;
END_ENTITY;


ENTITY joggle_termination
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SELF.description IN ['value termination','face termination'];
  wr3 : (SELF.description <> 'value termination') XOR (SIZEOF(QUERY(pdr <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) > 0);
  wr4 : (SELF.description <> 'value termination') OR (SIZEOF(QUERY(pdr <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(pdr.used_representation.items) = 2)
        )) > 0);
  wr5 : (SELF.description <> 'value termination') OR (SIZEOF(QUERY(pdr <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'depth') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1);
  wr6 : (SELF.description <> 'value termination') OR (SIZEOF(QUERY(pdr <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'angle') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) =
         2))) = 1))) = 1);
  wr7 : SIZEOF(QUERY(fcs <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcs)) AND
         (fcs.name = 'joggle termination usage') AND (
        'AUTOMOTIVE_DESIGN.JOGGLE' IN TYPEOF(fcs.related_shape_aspect.of_shape
        .definition)) AND (fcs.related_shape_aspect.name = 'termination') AND
```

```
        (fcs.related_shape_aspect.description =
        'joggle termination occurrence'))) = 1;
END_ENTITY;


ENTITY kinematic_analysis_result;
 analysed_mechanism : mechanism;
 contained_kinematic_results : SET[1:?] OF kinematic_result;
 WHERE
  wr1 : (SIZEOF(QUERY(result <* contained_kinematic_results |
        'AUTOMOTIVE_DESIGN.' + 'RESULTING_PATH' IN TYPEOF(result))) > 0) XOR (
        SIZEOF(QUERY(result <* contained_kinematic_results | ((
        'AUTOMOTIVE_DESIGN.' + 'INTERPOLATED_CONFIGURATION_SEQUENCE' IN TYPEOF
        (result)) AND (SIZEOF(QUERY(sequence <* result\
        interpolated_configuration_sequence.interpolation | (sequence.
        interpolation <> undefined))) > 0)))) = 0);
END_ENTITY;


ENTITY kinematic_frame_background_representation
 SUBTYPE OF (representation);
 SELF\representation.items : SET[1:?] OF kinematic_frame_background;
 SELF\representation.context_of_items : geometric_representation_context;
 WHERE
  wr1 : SELF\representation.context_of_items\geometric_representation_context.
        coordinate_space_dimension = 3;
END_ENTITY;


ENTITY kinematic_frame_background_representation_association
 SUBTYPE OF (representation_relationship_with_transformation);
 SELF\representation_relationship_with_transformation.transformation_operator
 : kinematic_frame_based_transformation;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.KINEMATIC_LINK_REPRESENTATION' IN TYPEOF(SELF\
        representation_relationship.rep_1);
  wr2 : 'AUTOMOTIVE_DESIGN.KINEMATIC_FRAME_BACKGROUND_REPRESENTATION' IN
        TYPEOF(SELF\representation_relationship.rep_2);
  wr3 : SELF\representation_relationship_with_transformation.
        transformation_operator\kinematic_frame_based_transformation.
        transformer_frame\representation_item IN SELF\
        representation_relationship.rep_1.items;
END_ENTITY;


ENTITY kinematic_frame_based_transformation
 SUBTYPE OF (functionally_defined_transformation);
 transformer_frame : rigid_placement;
END_ENTITY;


ENTITY kinematic_ground_representation
 SUBTYPE OF (representation);
 INVERSE
```

```
  property : kinematic_property_representation_relation FOR used_representation
             ;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_CONTEXT' IN TYPEOF(SELF\
        representation.context_of_items);
END_ENTITY;


ENTITY kinematic_joint;
 first_link : kinematic_link;
 second_link : kinematic_link;
 INVERSE
  structure : kinematic_structure FOR joints;
 WHERE
  wr1 : first_link :<>: second_link;
END_ENTITY;


ENTITY kinematic_link;
 WHERE
  wr1 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.KINEMATIC_JOINT.FIRST_LINK') +
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.KINEMATIC_JOINT.SECOND_LINK')) > 0;
  wr2 : unique_link_usage(SELF);
END_ENTITY;


ENTITY kinematic_link_representation
 SUBTYPE OF (representation);
 SELF\representation.context_of_items : geometric_representation_context;
 DERIVE
  link_frame : geometric_representation_context := SELF\representation.
               context_of_items;
 INVERSE
  link_representation_relation : kinematic_link_representation_relation FOR
                                 geometric_aspects;
 WHERE
  wr1 : SIZEOF(QUERY(item <* SELF\representation.items | NOT ((
        'AUTOMOTIVE_DESIGN.RIGID_PLACEMENT' IN TYPEOF(item)) OR (
        'AUTOMOTIVE_DESIGN.CARTESIAN_TRANSFORMATION_OPERATOR_3D' IN TYPEOF(
        item))))) = 0;
END_ENTITY;


ENTITY kinematic_link_representation_association
 SUBTYPE OF (representation_relationship);
 SELF\representation_relationship.rep_1 : kinematic_link_representation;
 WHERE
  wr1 : SELF\representation_relationship.rep_2.context_of_items :=: SELF\
        representation_relationship.rep_1\representation.context_of_items;
  wr2 : SIZEOF(['AUTOMOTIVE_DESIGN.KINEMATIC_GROUND_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.KINEMATIC_LINK_REPRESENTATION'] * TYPEOF(SELF\
        representation_relationship.rep_2)) = 0;
END_ENTITY;
```

```
ENTITY kinematic_link_representation_relation;
 topological_aspects : kinematic_link;
 geometric_aspects : kinematic_link_representation;
 UNIQUE
  ur1 : topological_aspects;
END_ENTITY;


ENTITY kinematic_pair
 SUBTYPE OF (item_defined_transformation);
 joint : kinematic_joint;
 DERIVE
  pair_placement_in_first_link_context : rigid_placement := SELF\
                                         item_defined_transformation.
                                         transform_item_1;
  pair_placement_in_second_link_context : rigid_placement := SELF\
                                          item_defined_transformation.
                                          transform_item_2;
 UNIQUE
  ur1 : joint;
 WHERE
  wr1 : coordinated_pair_link_representation(joint.first_link,
        pair_placement_in_first_link_context);
  wr2 : coordinated_pair_link_representation(joint.second_link,
        pair_placement_in_second_link_context);
END_ENTITY;


ENTITY kinematic_property_definition
 SUBTYPE OF (property_definition);
 ground_definition : characterized_definition;
END_ENTITY;


ENTITY kinematic_property_representation_relation
 SUBTYPE OF (property_definition_representation);
 UNIQUE
  ur1 : SELF\property_definition_representation.definition;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.KINEMATIC_PROPERTY_DEFINITION' IN TYPEOF(SELF\
        property_definition_representation.definition);
  wr2 : 'AUTOMOTIVE_DESIGN.KINEMATIC_GROUND_REPRESENTATION' IN TYPEOF(SELF\
        property_definition_representation.used_representation);
END_ENTITY;


ENTITY kinematic_structure;
 joints : SET[1:?] OF kinematic_joint;
END_ENTITY;


ENTITY known_source
 SUBTYPE OF (external_source,pre_defined_item);
 WHERE
```

```
  wr1 : SELF.name = 'ISO 13584 dictionary';
END_ENTITY;


ENTITY language_role;
 name : label;
 description : text;
END_ENTITY;


ENTITY leader_curve
 SUBTYPE OF (annotation_curve_occurrence);
 WHERE
  wr1 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'LEADER_DIRECTED_CALLOUT.CONTENTS')) >= 1;
END_ENTITY;


ENTITY leader_directed_callout
 SUBTYPE OF (draughting_callout);
 WHERE
  wr1 : SIZEOF(QUERY(l_1 <* SELF\draughting_callout.contents |
        'AUTOMOTIVE_DESIGN.LEADER_CURVE' IN (TYPEOF(l_1)))) >= 1;
  wr2 : SIZEOF(SELF\draughting_callout.contents) >= 2;
END_ENTITY;


ENTITY leader_directed_dimension
 SUBTYPE OF (leader_directed_callout);
 WHERE
  wr1 : SIZEOF(QUERY(con <* SELF.contents | 'AUTOMOTIVE_DESIGN.LEADER_CURVE'
        IN TYPEOF(con))) = 1;
END_ENTITY;


ENTITY leader_terminator
 SUBTYPE OF (terminator_symbol);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.LEADER_CURVE' IN TYPEOF(SELF\terminator_symbol.
        annotated_curve);
END_ENTITY;


ENTITY length_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.LENGTH_UNIT' IN TYPEOF(SELF\measure_with_unit.
        unit_component);
END_ENTITY;


ENTITY length_unit
 SUBTYPE OF (named_unit);
 WHERE
  wr1 : (SELF\named_unit.dimensions.length_exponent = 1.0) AND (SELF\
```

```
        named_unit.dimensions.mass_exponent = 0.0) AND (SELF\named_unit.
        dimensions.time_exponent = 0.0) AND (SELF\named_unit.dimensions.
        electric_current_exponent = 0.0) AND (SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 0.0) AND (SELF\named_unit.
        dimensions.amount_of_substance_exponent = 0.0) AND (SELF\named_unit.
        dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;


ENTITY light_source
 SUPERTYPE OF (ONEOF(light_source_ambient,light_source_directional,
              light_source_positional,light_source_spot))
 SUBTYPE OF (geometric_representation_item);
 light_colour : colour;
 WHERE
  wr1 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'STYLED_ITEM')) = 0;
END_ENTITY;


ENTITY light_source_ambient
 SUBTYPE OF (light_source);
END_ENTITY;


ENTITY light_source_directional
 SUBTYPE OF (light_source);
 orientation : direction;
END_ENTITY;


ENTITY light_source_positional
 SUBTYPE OF (light_source);
 position : cartesian_point;
 constant_attenuation : REAL;
 distance_attenuation : REAL;
END_ENTITY;


ENTITY light_source_spot
 SUBTYPE OF (light_source);
 position : cartesian_point;
 orientation : direction;
 concentration_exponent : REAL;
 constant_attenuation : REAL;
 distance_attenuation : REAL;
 spread_angle : positive_plane_angle_measure;
END_ENTITY;


ENTITY limits_and_fits;
 form_variance : label;
 zone_variance : label;
 grade : label;
 source : text;
```

```
END_ENTITY;


ENTITY line
 SUBTYPE OF (curve);
 pnt : cartesian_point;
 dir : vector;
 WHERE
  wr1 : dir.dim = pnt.dim;
END_ENTITY;


ENTITY linear_dimension
 SUBTYPE OF (dimension_curve_directed_callout);
END_ENTITY;


ENTITY linear_profile_tolerance
 SUBTYPE OF (geometric_tolerance);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT' IN
         TYPEOF(SELF));
  wr2 : (NOT (('AUTOMOTIVE_DESIGN.' +
        'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF))) OR (
        SIZEOF(SELF\geometric_tolerance_with_datum_reference.datum_system) <=
        3);
END_ENTITY;


ENTITY local_time;
 hour_component : hour_in_day;
 minute_component : OPTIONAL minute_in_hour;
 second_component : OPTIONAL second_in_minute;
 zone : coordinated_universal_time_offset;
 WHERE
  wr1 : valid_time(SELF);
END_ENTITY;


ENTITY location_shape_representation
 SUBTYPE OF (representation);
 WHERE
  wr1 : SIZEOF(SELF.items) = 1;
  wr2 : SIZEOF(QUERY(i <* SELF.items | 'AUTOMOTIVE_DESIGN.POINT' IN TYPEOF(i))
        ) = 1;
END_ENTITY;


ENTITY locator
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
```

```
        ) | SIZEOF(pdr.used_representation.items) = 4)) = 1)) = 1;
  wr2 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'diameter') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1)) = 1;
  wr3 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'tip radius') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1)) = 1;
  wr4 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'tip angle') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) =
         2))) = 1))) = 1)) = 1;
END_ENTITY;


ENTITY loop
 SUPERTYPE OF (ONEOF(vertex_loop,edge_loop,poly_loop))
 SUBTYPE OF (topological_representation_item);
END_ENTITY;


ENTITY lot_effectivity
 SUBTYPE OF (effectivity);
 effectivity_lot_id : identifier;
 effectivity_lot_size : measure_with_unit;
END_ENTITY;


ENTITY luminous_intensity_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.LUMINOUS_INTENSITY_UNIT' IN TYPEOF(SELF\
        measure_with_unit.unit_component);
END_ENTITY;


ENTITY luminous_intensity_unit
 SUBTYPE OF (named_unit);
```

```
 WHERE
  wr1 : (SELF\named_unit.dimensions.length_exponent = 0.0) AND (SELF\
        named_unit.dimensions.mass_exponent = 0.0) AND (SELF\named_unit.
        dimensions.time_exponent = 0.0) AND (SELF\named_unit.dimensions.
        electric_current_exponent = 0.0) AND (SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 0.0) AND (SELF\named_unit.
        dimensions.amount_of_substance_exponent = 0.0) AND (SELF\named_unit.
        dimensions.luminous_intensity_exponent = 1.0);
END_ENTITY;


ENTITY make_from_usage_option
 SUBTYPE OF (product_definition_usage);
 ranking : INTEGER;
 ranking_rationale : text;
 quantity : measure_with_unit;
 WHERE
  wr1 : ranking > 0;
END_ENTITY;


ENTITY manifold_solid_brep
 SUBTYPE OF (solid_model);
 outer : closed_shell;
END_ENTITY;


ENTITY manifold_surface_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(it <* SELF.items | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM','AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D'
        ] * TYPEOF(it)) = 1))) = 0;
  wr2 : SIZEOF(QUERY(it <* SELF.items | SIZEOF([
        'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM'] * TYPEOF(it)) = 1)) > 0;
  wr3 : SIZEOF(QUERY(mi <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(it)) | NOT (
        'AUTOMOTIVE_DESIGN.MANIFOLD_SURFACE_SHAPE_REPRESENTATION' IN TYPEOF(mi
        \mapped_item.mapping_source.mapped_representation)))) = 0;
  wr4 : SIZEOF(QUERY(sbsm <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
        SIZEOF(QUERY(sh <* sbsm\shell_based_surface_model.sbsm_boundary | NOT (
        SIZEOF(['AUTOMOTIVE_DESIGN.OPEN_SHELL',
        'AUTOMOTIVE_DESIGN.CLOSED_SHELL'] * TYPEOF(sh)) = 1))) = 0))) = 0;
  wr5 : SIZEOF(QUERY(sbsm <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
        SIZEOF(QUERY(cfs <* sbsm\shell_based_surface_model.sbsm_boundary | NOT
        (SIZEOF(QUERY(fa <* cfs\connected_face_set.cfs_faces | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.FACE_SURFACE','AUTOMOTIVE_DESIGN.ORIENTED_FACE'] *
        TYPEOF(fa)) = 1))) = 0))) = 0))) = 0;
  wr6 : SIZEOF(QUERY(sbsm <* QUERY(it <* SELF.items |
```

```
        'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
        SIZEOF(QUERY(cfs <* sbsm\shell_based_surface_model.sbsm_boundary | NOT
        (SIZEOF(QUERY(f_sf <* QUERY(fa <* cfs\connected_face_set.cfs_faces |
        'AUTOMOTIVE_DESIGN.FACE_SURFACE' IN TYPEOF(fa)) | NOT ((
        'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(f_sf)) OR (SIZEOF([
        'AUTOMOTIVE_DESIGN.OFFSET_SURFACE','AUTOMOTIVE_DESIGN.SURFACE_REPLICA'
        ] * TYPEOF(f_sf\face_surface.face_geometry)) = 1)))) = 0))) = 0))) = 0
        ;
wr7 : SIZEOF(QUERY(sbsm <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
        SIZEOF(QUERY(cfs <* sbsm\shell_based_surface_model.sbsm_boundary | NOT
        (SIZEOF(QUERY(fa <* cfs\connected_face_set.cfs_faces | NOT ((
        'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (
        basis_surface_check(fa\face_surface.face_geometry))))) = 0))) = 0))) =
         0;
wr8 : SIZEOF(QUERY(sbsm <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
        SIZEOF(QUERY(cfs <* sbsm\shell_based_surface_model.sbsm_boundary | NOT
        (SIZEOF(QUERY(fa <* cfs\connected_face_set.cfs_faces | NOT ((
        'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(bnds
        <* fa.bounds | NOT (SIZEOF(['AUTOMOTIVE_DESIGN.EDGE_LOOP',
        'AUTOMOTIVE_DESIGN.VERTEX_LOOP'] * TYPEOF(bnds.bound)) = 1))) = 0))))
        = 0))) = 0))) = 0;
wr9 : SIZEOF(QUERY(sbsm <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
        SIZEOF(QUERY(cfs <* sbsm\shell_based_surface_model.sbsm_boundary | NOT
        (SIZEOF(QUERY(fa <* cfs\connected_face_set.cfs_faces | NOT ((
        'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
        elp_fbnds <* QUERY(bnds <* fa.bounds | 'AUTOMOTIVE_DESIGN.EDGE_LOOP'
        IN TYPEOF(bnds)) | NOT (SIZEOF(QUERY(oe <* elp_fbnds.bound\path.
        edge_list | NOT ('AUTOMOTIVE_DESIGN.EDGE_CURVE' IN TYPEOF(oe.
        edge_element)))) = 0))) = 0)))) = 0))) = 0))) = 0;
wr10 : SIZEOF(QUERY(sbsm <* QUERY(it <* SELF.items |
         'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
         SIZEOF(QUERY(cfs <* sbsm\shell_based_surface_model.sbsm_boundary |
         NOT (SIZEOF(QUERY(fa <* cfs\connected_face_set.cfs_faces | NOT ((
         'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
         elp_fbnds <* QUERY(bnds <* fa.bounds | 'AUTOMOTIVE_DESIGN.EDGE_LOOP'
         IN TYPEOF(bnds.bound)) | NOT (SIZEOF(QUERY(oe_cv <* QUERY(oe <*
         elp_fbnds.bound\path.edge_list | 'AUTOMOTIVE_DESIGN.EDGE_CURVE' IN
         TYPEOF(oe.edge_element)) | NOT (SIZEOF([
         'AUTOMOTIVE_DESIGN.CURVE_REPLICA','AUTOMOTIVE_DESIGN.OFFSET_CURVE_3D'
         ,'AUTOMOTIVE_DESIGN.PCURVE'] * TYPEOF(oe_cv.edge_element\edge_curve.
         edge_geometry)) = 1))) = 0))) = 0)))) = 0))) = 0))) = 0;
wr11 : SIZEOF(QUERY(sbsm <* QUERY(it <* SELF.items |
         'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
         SIZEOF(QUERY(cfs <* sbsm\shell_based_surface_model.sbsm_boundary |
         NOT (SIZEOF(QUERY(fa <* cfs\connected_face_set.cfs_faces | NOT ((
         'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
         elp_fbnds <* QUERY(bnds <* fa.bounds | 'AUTOMOTIVE_DESIGN.EDGE_LOOP'
         IN TYPEOF(bnds.bound)) | NOT (SIZEOF(QUERY(oe <* elp_fbnds.bound\path
         .edge_list | NOT (basis_curve_check(oe.edge_element\edge_curve.
```

```
       edge_geometry)))) = 0))) = 0)))) = 0))) = 0))) = 0;
wr12 : SIZEOF(QUERY(sbsm <* QUERY(it <* SELF.items |
       'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
       SIZEOF(QUERY(cfs <* sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF(QUERY(fa <* cfs\connected_face_set.cfs_faces | NOT ((
       'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
       elp_fbnds <* QUERY(bnds <* fa.bounds | 'AUTOMOTIVE_DESIGN.EDGE_LOOP'
       IN TYPEOF(bnds)) | NOT (SIZEOF(QUERY(oe <* elp_fbnds.bound\path.
       edge_list | NOT (('AUTOMOTIVE_DESIGN.VERTEX_POINT' IN TYPEOF(oe.
       edge_element.edge_start)) AND ('AUTOMOTIVE_DESIGN.VERTEX_POINT' IN
       TYPEOF(oe.edge_element.edge_end))))) = 0))) = 0)))) = 0))) = 0))) = 0
       ;
wr13 : SIZEOF(QUERY(sbsm <* QUERY(it <* SELF.items |
       'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
       SIZEOF(QUERY(cfs <* sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF(QUERY(fa <* cfs\connected_face_set.cfs_faces | NOT ((
       'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
       elp_fbnds <* QUERY(bnds <* fa.bounds | 'AUTOMOTIVE_DESIGN.EDGE_LOOP'
       IN TYPEOF(bnds.bound)) | NOT (SIZEOF(QUERY(oe <* elp_fbnds.bound\path
       .edge_list | NOT ((SIZEOF(['AUTOMOTIVE_DESIGN.CARTESIAN_POINT',
       'AUTOMOTIVE_DESIGN.DEGENERATE_PCURVE',
       'AUTOMOTIVE_DESIGN.POINT_ON_CURVE',
       'AUTOMOTIVE_DESIGN.POINT_ON_SURFACE'] * TYPEOF(oe.edge_element.
       edge_start\vertex_point.vertex_geometry)) = 1) AND (SIZEOF([
       'AUTOMOTIVE_DESIGN.CARTESIAN_POINT',
       'AUTOMOTIVE_DESIGN.DEGENERATE_PCURVE',
       'AUTOMOTIVE_DESIGN.POINT_ON_CURVE',
       'AUTOMOTIVE_DESIGN.POINT_ON_SURFACE'] * TYPEOF(oe.edge_element.
       edge_end\vertex_point.vertex_geometry)) = 1)))) = 0))) = 0)))) = 0)))
        = 0))) = 0;
wr14 : SIZEOF(QUERY(sbsm <* QUERY(it <* SELF.items |
       'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
       SIZEOF(QUERY(cfs <* sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF(QUERY(fa <* cfs\connected_face_set.cfs_faces | NOT ((
       'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
       vlp_fbnds <* QUERY(bnds <* fa.bounds |
       'AUTOMOTIVE_DESIGN.VERTEX_LOOP' IN TYPEOF(bnds.bound)) | NOT (
       'AUTOMOTIVE_DESIGN.VERTEX_POINT' IN TYPEOF(vlp_fbnds.bound\
       vertex_loop.loop_vertex)))) = 0)))) = 0))) = 0))) = 0;
wr15 : SIZEOF(QUERY(sbsm <* QUERY(it <* SELF.items |
       'AUTOMOTIVE_DESIGN.SHELL_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
       SIZEOF(QUERY(cfs <* sbsm\shell_based_surface_model.sbsm_boundary |
       NOT (SIZEOF(QUERY(fa <* cfs\connected_face_set.cfs_faces | NOT ((
       'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
       vlp_fbnds <* QUERY(bnds <* fa.bounds |
       'AUTOMOTIVE_DESIGN.VERTEX_LOOP' IN TYPEOF(bnds.bound)) | NOT (SIZEOF(
       ['AUTOMOTIVE_DESIGN.CARTESIAN_POINT',
       'AUTOMOTIVE_DESIGN.DEGENERATE_PCURVE',
       'AUTOMOTIVE_DESIGN.POINT_ON_CURVE',
       'AUTOMOTIVE_DESIGN.POINT_ON_SURFACE'] * TYPEOF(vlp_fbnds.bound\
       vertex_loop.loop_vertex\vertex_point.vertex_geometry)) = 1))) = 0))))
        = 0))) = 0))) = 0;
```

```
END_ENTITY;


ENTITY mapped_item
 SUBTYPE OF (representation_item);
 mapping_source : representation_map;
 mapping_target : representation_item;
 WHERE
  wr1 : acyclic_mapped_representation(using_representations(SELF),[SELF]);
END_ENTITY;


ENTITY mass_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.MASS_UNIT' IN TYPEOF(SELF\measure_with_unit.
        unit_component);
END_ENTITY;


ENTITY mass_unit
 SUBTYPE OF (named_unit);
 WHERE
  wr1 : (SELF\named_unit.dimensions.length_exponent = 0.0) AND (SELF\
        named_unit.dimensions.mass_exponent = 1.0) AND (SELF\named_unit.
        dimensions.time_exponent = 0.0) AND (SELF\named_unit.dimensions.
        electric_current_exponent = 0.0) AND (SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 0.0) AND (SELF\named_unit.
        dimensions.amount_of_substance_exponent = 0.0) AND (SELF\named_unit.
        dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;


ENTITY mated_item_relationship
 SUBTYPE OF (product_definition,product_definition_relationship);
 WHERE
  wr1 : (SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.' + 'RELATED_PRODUCT_DEFINITION')) =
        1) AND (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.' + 'RELATED_PRODUCT_DEFINITION') | (
        pdr.name = 'mating context') AND (SIZEOF(QUERY(gma <* USEDIN(pdr.
        relating_product_definition,
        'AUTOMOTIVE_DESIGN.GROUP_MEMBER_ASSIGNMENT.ITEMS') | (gma.
        assigned_group.name = 'mating definition') AND (gma.role.name =
        'mating'))) = 1))) = 1);
  wr2 : ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_OCCURRENCE' IN TYPEOF(SELF.
        relating_product_definition)) AND (SIZEOF(QUERY(pdr <* USEDIN(SELF.
        relating_product_definition,
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RELATIONSHIP.' +
        'RELATED_PRODUCT_DEFINITION') | (pdr.name = 'mating membership') AND (
        SIZEOF(QUERY(pdr1 <* USEDIN(pdr.relating_product_definition,
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RELATIONSHIP.' +
        'RELATING_PRODUCT_DEFINITION') | (pdr1.name = 'mating context') AND (
        pdr1.related_product_definition :=: SELF))) = 1))) = 1);
```

```
  wr3 : ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_OCCURRENCE' IN TYPEOF(SELF.
        related_product_definition)) AND (SIZEOF(QUERY(pdr <* USEDIN(SELF.
        related_product_definition,
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RELATIONSHIP.' +
        'RELATED_PRODUCT_DEFINITION') | (pdr.name = 'mating membership') AND (
        SIZEOF(QUERY(pdr1 <* USEDIN(pdr.relating_product_definition,
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RELATIONSHIP.' +
        'RELATING_PRODUCT_DEFINITION') | (pdr1.name = 'mating context') AND (
        pdr1.related_product_definition :=: SELF))) = 1))) = 1);
END_ENTITY;


ENTITY material_designation;
 name : label;
 definitions : SET[1:?] OF characterized_definition;
END_ENTITY;


ENTITY material_designation_characterization;
 name : label;
 description : text;
 designation : material_designation;
 property : characterized_material_property;
END_ENTITY;


ENTITY material_property
 SUBTYPE OF (property_definition);
 UNIQUE
  ur1 : SELF\property_definition.name,SELF\property_definition.definition;
 WHERE
  wr1 : ('AUTOMOTIVE_DESIGN.CHARACTERIZED_OBJECT' IN TYPEOF(SELF\
        property_definition.definition)) OR (SIZEOF(bag_to_set(USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        )) - QUERY(temp <* bag_to_set(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) | (
        'AUTOMOTIVE_DESIGN.' + 'MATERIAL_PROPERTY_REPRESENTATION' IN TYPEOF(
        temp)))) = 0);
END_ENTITY;


ENTITY material_property_representation
 SUBTYPE OF (property_definition_representation);
 dependent_environment : data_environment;
END_ENTITY;


ENTITY measure_qualification;
 name : label;
 description : text;
 qualified_measure : measure_with_unit;
 qualifiers : SET[1:?] OF value_qualifier;
 WHERE
  wr1 : SIZEOF(QUERY(temp <* qualifiers |
```

```
        'AUTOMOTIVE_DESIGN.PRECISION_QUALIFIER' IN TYPEOF(temp))) < 2;
END_ENTITY;


ENTITY measure_representation_item
 SUBTYPE OF (representation_item,measure_with_unit);
END_ENTITY;


ENTITY measure_with_unit
 SUPERTYPE OF (ONEOF(length_measure_with_unit,mass_measure_with_unit,
               time_measure_with_unit,electric_current_measure_with_unit,
               thermodynamic_temperature_measure_with_unit,
               amount_of_substance_measure_with_unit,
               luminous_intensity_measure_with_unit,
               plane_angle_measure_with_unit,solid_angle_measure_with_unit,
               area_measure_with_unit,volume_measure_with_unit,
               ratio_measure_with_unit));
 value_component : measure_value;
 unit_component : unit;
 WHERE
  wr1 : valid_units(SELF);
END_ENTITY;


ENTITY mechanical_design_geometric_presentation_area
 SUBTYPE OF (presentation_area);
 UNIQUE
  ur1 : SELF\representation.name;
 WHERE
  wr1 : SIZEOF(QUERY(it1 <* SELF.items | NOT ((
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT' IN TYPEOF(it1)) OR ((
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(it1)) AND (
        'AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN TYPEOF(it1\mapped_item.
        mapping_source.mapped_representation)))))) = 0;
  wr2 : SIZEOF(QUERY(pv <* QUERY(mi1 <* QUERY(it1 <* SELF.items |
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(it1)) |
        'AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN TYPEOF(mi1\mapped_item.
        mapping_source.mapped_representation)) | NOT (SIZEOF(QUERY(it2 <* pv\
        mapped_item.mapping_source.mapped_representation\presentation_view.
        items | NOT (('AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT' IN TYPEOF(it2)) OR (
        ('AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(it2)) AND NOT (
        'AUTOMOTIVE_DESIGN.' + 'CAMERA_IMAGE_3D_WITH_SCALE' IN TYPEOF(it2)))
        AND NOT ('AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN TYPEOF(it2\
        mapped_item.mapping_source.mapped_representation))) OR ((
        'AUTOMOTIVE_DESIGN.' + 'CAMERA_IMAGE_3D_WITH_SCALE' IN TYPEOF(it2))
        AND NOT (('AUTOMOTIVE_DESIGN.' +
        'MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION' IN TYPEOF(
        it2\mapped_item.mapping_source.mapped_representation)))))) = 0))) = 0;
END_ENTITY;


ENTITY mechanical_design_geometric_presentation_representation
 SUBTYPE OF (representation);
```

```
UNIQUE
 ur1 : SELF\representation.name;
WHERE
 wr1 : SIZEOF(QUERY(it <* SELF.items | NOT (SIZEOF([
       'AUTOMOTIVE_DESIGN.MAPPED_ITEM','AUTOMOTIVE_DESIGN.STYLED_ITEM',
       'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT',
       'AUTOMOTIVE_DESIGN.CAMERA_MODEL_D3'] * TYPEOF(it)) = 1))) = 0;
 wr2 : SIZEOF(QUERY(mi <* QUERY(it <* SELF.items | (
       'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(it))) | NOT (SIZEOF([
       'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION','AUTOMOTIVE_DESIGN.' +
       'MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION'] * TYPEOF(mi
       \mapped_item.mapping_source.mapped_representation)) = 1))) = 0;
 wr3 : SIZEOF(QUERY(smi <* QUERY(si <* QUERY(it <* SELF.items | (
       'AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it))) | (
       'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(si\styled_item.item))) | NOT
       (('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN TYPEOF(smi\styled_item.
       item\mapped_item.mapping_source.mapped_representation))))) = 0;
 wr4 : SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
       'AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa
       <* si\styled_item.styles | NOT (SIZEOF(QUERY(pss <* psa.styles | NOT (
       SIZEOF(['AUTOMOTIVE_DESIGN.POINT_STYLE',
       'AUTOMOTIVE_DESIGN.CURVE_STYLE',
       'AUTOMOTIVE_DESIGN.SURFACE_STYLE_USAGE'] * TYPEOF(pss)) = 1))) = 0)))
       = 0))) = 0;
 wr5 : SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
       'AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(
       psbc <* QUERY(psa <* si\styled_item.styles | 'AUTOMOTIVE_DESIGN.' +
       'PRESENTATION_STYLE_BY_CONTEXT' IN TYPEOF(psa)) | NOT (SIZEOF([
       'AUTOMOTIVE_DESIGN.REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.REPRESENTATION'] * TYPEOF(psbc\
       presentation_style_by_context.style_context)) = 1))) = 0))) = 0;
 wr6 : SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
       'AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa
       <* si\styled_item.styles | NOT (SIZEOF(QUERY(ps <* QUERY(pss <* psa.
       styles | 'AUTOMOTIVE_DESIGN.POINT_STYLE' IN TYPEOF(pss)) | NOT ((
       'AUTOMOTIVE_DESIGN.MARKER_TYPE' IN TYPEOF(ps\point_style.marker)) AND
       ('AUTOMOTIVE_DESIGN.' + 'POSITIVE_LENGTH_MEASURE' IN TYPEOF(ps\
       point_style.marker_size)) AND (SIZEOF(['AUTOMOTIVE_DESIGN.COLOUR_RGB',
       'AUTOMOTIVE_DESIGN.' + 'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(ps\
       point_style.marker_colour)) = 1)))) = 0))) = 0))) = 0;
 wr7 : SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
       'AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa
       <* si\styled_item.styles | NOT (SIZEOF(QUERY(cs <* QUERY(pss <* psa.
       styles | 'AUTOMOTIVE_DESIGN.CURVE_STYLE' IN TYPEOF(pss)) | NOT ((
       SIZEOF(['AUTOMOTIVE_DESIGN.COLOUR_RGB','AUTOMOTIVE_DESIGN.' +
       'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(cs\curve_style.curve_colour)
       ) = 1) AND ('AUTOMOTIVE_DESIGN.' + 'POSITIVE_LENGTH_MEASURE' IN TYPEOF
       (cs\curve_style.curve_width)) AND (SIZEOF([
       'AUTOMOTIVE_DESIGN.CURVE_STYLE_FONT','AUTOMOTIVE_DESIGN.' +
       'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(cs\curve_style.
       curve_font)) = 1)))) = 0))) = 0))) = 0;
 wr8 : SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
```

```
        'AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa
        <* si\styled_item.styles | NOT (SIZEOF(QUERY(ssu <* QUERY(pss <* psa.
        styles | 'AUTOMOTIVE_DESIGN.SURFACE_STYLE_USAGE' IN TYPEOF(pss)) | NOT
        ('AUTOMOTIVE_DESIGN.' + 'SURFACE_SIDE_STYLE' IN TYPEOF(ssu\
        surface_style_usage.style)))) = 0))) = 0))) = 0;
  wr9 : SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(psa
        <* si\styled_item.styles | NOT (SIZEOF(QUERY(ssu <* QUERY(pss <* psa.
        styles | 'AUTOMOTIVE_DESIGN.SURFACE_STYLE_USAGE' IN TYPEOF(pss)) | NOT
        (SIZEOF(QUERY(sses <* ssu\surface_style_usage.style\surface_side_style
        .styles | NOT (SIZEOF(['AUTOMOTIVE_DESIGN.' +
        'SURFACE_STYLE_PARAMETER_LINE','AUTOMOTIVE_DESIGN.' +
        'SURFACE_STYLE_CONTROL_GRID','AUTOMOTIVE_DESIGN.' +
        'SURFACE_STYLE_SILHOUETTE','AUTOMOTIVE_DESIGN.' +
        'SURFACE_STYLE_SEGMENTATION_CURVE','AUTOMOTIVE_DESIGN.' +
        'SURFACE_STYLE_BOUNDARY'] * TYPEOF(sses)) = 1))) = 0))) = 0))) = 0)))
        = 0;
 wr10 : SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
         'AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(
         psa <* si\styled_item.styles | NOT (SIZEOF(QUERY(ssu <* QUERY(pss <*
         psa.styles | 'AUTOMOTIVE_DESIGN.SURFACE_STYLE_USAGE' IN TYPEOF(pss))
          | NOT (SIZEOF(QUERY(sspl <* QUERY(sses <* ssu\surface_style_usage.
         style\surface_side_style.styles | 'AUTOMOTIVE_DESIGN.' +
         'SURFACE_STYLE_PARAMETER_LINE' IN TYPEOF(sses)) | NOT ((
         'AUTOMOTIVE_DESIGN.CURVE_STYLE' IN TYPEOF(sspl\
         surface_style_parameter_line.style_of_parameter_lines)) AND (SIZEOF([
         'AUTOMOTIVE_DESIGN.COLOUR_RGB','AUTOMOTIVE_DESIGN.' +
         'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sspl\
         surface_style_parameter_line.style_of_parameter_lines\curve_style.
         curve_colour)) = 1) AND ('AUTOMOTIVE_DESIGN.' +
         'POSITIVE_LENGTH_MEASURE' IN TYPEOF(sspl\surface_style_parameter_line
         .style_of_parameter_lines\curve_style.curve_width)) AND (SIZEOF([
         'AUTOMOTIVE_DESIGN.CURVE_STYLE_FONT','AUTOMOTIVE_DESIGN.' +
         'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(sspl\
         surface_style_parameter_line.style_of_parameter_lines\curve_style.
         curve_font)) = 1)))) = 0))) = 0))) = 0))) = 0;
 wr11 : SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
         'AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(
         psa <* si\styled_item.styles | NOT (SIZEOF(QUERY(ssu <* QUERY(pss <*
         psa.styles | 'AUTOMOTIVE_DESIGN.SURFACE_STYLE_USAGE' IN TYPEOF(pss))
          | NOT (SIZEOF(QUERY(sscg <* QUERY(sses <* ssu\surface_style_usage.
         style\surface_side_style.styles | 'AUTOMOTIVE_DESIGN.' +
         'SURFACE_STYLE_CONTROL_GRID' IN TYPEOF(sses)) | NOT ((
         'AUTOMOTIVE_DESIGN.CURVE_STYLE' IN TYPEOF(sscg\
         surface_style_control_grid.style_of_control_grid)) AND (SIZEOF([
         'AUTOMOTIVE_DESIGN.COLOUR_RGB','AUTOMOTIVE_DESIGN.' +
         'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sscg\
         surface_style_control_grid.style_of_control_grid\curve_style.
         curve_colour)) = 1) AND ('AUTOMOTIVE_DESIGN.' +
         'POSITIVE_LENGTH_MEASURE' IN TYPEOF(sscg\surface_style_control_grid.
         style_of_control_grid\curve_style.curve_width)) AND (SIZEOF([
         'AUTOMOTIVE_DESIGN.CURVE_STYLE_FONT','AUTOMOTIVE_DESIGN.' +
```

```
        'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(sscg\
        surface_style_control_grid.style_of_control_grid\curve_style.
        curve_font)) = 1)))) = 0))) = 0))) = 0))) = 0;
wr12 : SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(
        psa <* si\styled_item.styles | NOT (SIZEOF(QUERY(ssu <* QUERY(pss <*
        psa.styles | 'AUTOMOTIVE_DESIGN.SURFACE_STYLE_USAGE' IN TYPEOF(pss))
         | NOT (SIZEOF(QUERY(sssh <* QUERY(sses <* ssu\surface_style_usage.
        style\surface_side_style.styles | 'AUTOMOTIVE_DESIGN.' +
        'SURFACE_STYLE_SILHOUETTE' IN TYPEOF(sses)) | NOT ((
        'AUTOMOTIVE_DESIGN.CURVE_STYLE' IN TYPEOF(sssh\
        surface_style_silhouette.style_of_silhouette)) AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.COLOUR_RGB','AUTOMOTIVE_DESIGN.' +
        'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sssh\
        surface_style_silhouette.style_of_silhouette\curve_style.curve_colour
        )) = 1) AND ('AUTOMOTIVE_DESIGN.' + 'POSITIVE_LENGTH_MEASURE' IN
        TYPEOF(sssh\surface_style_silhouette.style_of_silhouette\curve_style.
        curve_width)) AND (SIZEOF(['AUTOMOTIVE_DESIGN.CURVE_STYLE_FONT',
        'AUTOMOTIVE_DESIGN.' + 'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(
        sssh\surface_style_silhouette.style_of_silhouette\curve_style.
        curve_font)) = 1)))) = 0))) = 0))) = 0))) = 0;
wr13 : SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(
        psa <* si\styled_item.styles | NOT (SIZEOF(QUERY(ssu <* QUERY(pss <*
        psa.styles | 'AUTOMOTIVE_DESIGN.SURFACE_STYLE_USAGE' IN TYPEOF(pss))
         | NOT (SIZEOF(QUERY(sssc <* QUERY(sses <* ssu\surface_style_usage.
        style\surface_side_style.styles | 'AUTOMOTIVE_DESIGN.' +
        'SURFACE_STYLE_SEGMENTATION_CURVE' IN TYPEOF(sses)) | NOT ((
        'AUTOMOTIVE_DESIGN.CURVE_STYLE' IN TYPEOF(sssc\
        surface_style_segmentation_curve.style_of_segmentation_curve)) AND (
        SIZEOF(['AUTOMOTIVE_DESIGN.COLOUR_RGB','AUTOMOTIVE_DESIGN.' +
        'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(sssc\
        surface_style_segmentation_curve.style_of_segmentation_curve\
        curve_style.curve_colour)) = 1) AND ('AUTOMOTIVE_DESIGN.' +
        'POSITIVE_LENGTH_MEASURE' IN TYPEOF(sssc\
        surface_style_segmentation_curve.style_of_segmentation_curve\
        curve_style.curve_width)) AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.CURVE_STYLE_FONT','AUTOMOTIVE_DESIGN.' +
        'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(sssc\
        surface_style_segmentation_curve.style_of_segmentation_curve\
        curve_style.curve_font)) = 1)))) = 0))) = 0))) = 0))) = 0;
wr14 : SIZEOF(QUERY(si <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.STYLED_ITEM' IN TYPEOF(it)) | NOT (SIZEOF(QUERY(
        psa <* si\styled_item.styles | NOT (SIZEOF(QUERY(ssu <* QUERY(pss <*
        psa.styles | 'AUTOMOTIVE_DESIGN.SURFACE_STYLE_USAGE' IN TYPEOF(pss))
         | NOT (SIZEOF(QUERY(ssbd <* QUERY(sses <* ssu\surface_style_usage.
        style\surface_side_style.styles |
        'AUTOMOTIVE_DESIGN.SURFACE_STYLE_BOUNDARY' IN TYPEOF(sses)) | NOT ((
        'AUTOMOTIVE_DESIGN.CURVE_STYLE' IN TYPEOF(ssbd\surface_style_boundary
        .style_of_boundary)) AND (SIZEOF(['AUTOMOTIVE_DESIGN.COLOUR_RGB',
        'AUTOMOTIVE_DESIGN.' + 'DRAUGHTING_PRE_DEFINED_COLOUR'] * TYPEOF(ssbd
        \surface_style_boundary.style_of_boundary\curve_style.curve_colour))
```

```
         = 1) AND ('AUTOMOTIVE_DESIGN.' + 'POSITIVE_LENGTH_MEASURE' IN TYPEOF(
         ssbd\surface_style_boundary.style_of_boundary\curve_style.curve_width
         )) AND (SIZEOF(['AUTOMOTIVE_DESIGN.CURVE_STYLE_FONT',
         'AUTOMOTIVE_DESIGN.' + 'DRAUGHTING_PRE_DEFINED_CURVE_FONT'] * TYPEOF(
         ssbd\surface_style_boundary.style_of_boundary\curve_style.curve_font)
         ) = 1)))) = 0))) = 0))) = 0))) = 0;
END_ENTITY;


ENTITY mechanism;
 structure_definition : kinematic_structure;
 base : kinematic_link;
 containing_property : kinematic_property_definition;
 WHERE
  wr1 : SIZEOF(QUERY(joint <* structure_definition.joints | (base :=: joint.
        first_link) OR (base :=: joint.second_link))) > 0;
END_ENTITY;


ENTITY mechanism_base_placement
 SUBTYPE OF (representation_relationship_with_transformation);
 base_of_mechanism : mechanism;
 SELF\representation_relationship_with_transformation.transformation_operator
 : cartesian_transformation_operator_3d;
 DERIVE
  SELF\representation_relationship.rep_2 : kinematic_link_representation :=
                                           representation_of_link(
                                           base_of_mechanism.base);
 UNIQUE
  ur1 : base_of_mechanism;
 WHERE
  wr1 : ('AUTOMOTIVE_DESIGN.KINEMATIC_GROUND_REPRESENTATION' IN TYPEOF(SELF\
        representation_relationship.rep_1)) OR (
        'AUTOMOTIVE_DESIGN.KINEMATIC_LINK_REPRESENTATION' IN TYPEOF(SELF\
        representation_relationship.rep_1));
  wr2 : suitably_based_mechanism(SELF,base_of_mechanism);
  wr3 : SELF\representation_relationship_with_transformation.
        transformation_operator\representation_item IN SELF\
        representation_relationship.rep_1.items;
END_ENTITY;


ENTITY model_change
 SUBTYPE OF (property_definition);
 WHERE
  wr1 : SIZEOF(['AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RELATIONSHIP',
        'AUTOMOTIVE_DESIGN.SHAPE_ASPECT_RELATIONSHIP'] * TYPEOF(SELF.
        definition)) = 1;
  wr2 : (SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) <= 4) AND (SIZEOF(
        QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | NOT (pdr.name IN [
        'added elements','deleted elements','relating change description',
```

```
        'related change description']))) = 0);
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.name =
        'added elements')) <= 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.name =
        'deleted elements')) <= 1;
  wr5 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.name =
        'relating change description')) = 1;
  wr6 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.name =
        'related change description')) = 1;
  wr7 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.name =
        'relating change description'))) = 0) OR (SIZEOF(QUERY(pdr <* USEDIN(
        SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.name =
        'relating change description') AND (SIZEOF(QUERY(rr <* USEDIN(pdr.
        used_representation,
        'AUTOMOTIVE_DESIGN.REPRESENTATION_RELATIONSHIP.REP_1') | (SIZEOF(QUERY
        (pdr_1 <* USEDIN(rr.rep_2,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.' + 'USED_REPRESENTATION') | pdr_1
        .name = 'related change description')) = 1) AND (rr.name =
        'change description'))) = 1))) = 1);
END_ENTITY;


ENTITY modified_geometric_tolerance
 SUBTYPE OF (geometric_tolerance);
 modifier : limit_condition;
END_ENTITY;


ENTITY modified_pattern
 SUBTYPE OF (replicate_feature);
 WHERE
  wr1 : SIZEOF(QUERY(cor <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'CHARACTERIZED_OBJECT_RELATIONSHIP.RELATED_OBJECT') | (cor.name =
        'replicate base assignment') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.CIRCULAR_PATTERN',
        'AUTOMOTIVE_DESIGN.RECTANGULAR_PATTERN'] * TYPEOF(cor.relating_object)
        ) = 1))) = 1;
  wr2 : (SIZEOF(QUERY(cor <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'CHARACTERIZED_OBJECT_RELATIONSHIP.RELATED_OBJECT') | (cor.name =
        'replicate base assignment') AND ('AUTOMOTIVE_DESIGN.CIRCULAR_PATTERN'
         IN TYPEOF(cor.relating_object)))) = 0) OR (SIZEOF(QUERY(pds <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | NOT (sa.description IN ['circular omit',
        'circular offset']))) > 0))) = 0);
  wr3 : (SIZEOF(QUERY(cor <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
```

```
        'CHARACTERIZED_OBJECT_RELATIONSHIP.RELATED_OBJECT') | (cor.name =
        'replicate base assignment') AND ('AUTOMOTIVE_DESIGN.CIRCULAR_PATTERN'
         IN TYPEOF(cor.relating_object)))) = 0) OR (SIZEOF(QUERY(pds <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | sa.description IN ['circular omit',
        'circular offset'])) > 0))) >= 1);
wr4 : (SIZEOF(QUERY(cor <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'CHARACTERIZED_OBJECT_RELATIONSHIP.RELATED_OBJECT') | (cor.name =
        'replicate base assignment') AND (
        'AUTOMOTIVE_DESIGN.RECTANGULAR_PATTERN' IN TYPEOF(cor.relating_object)
        ))) = 0) OR (SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | NOT (sa.description IN ['rectangular omit',
        'rectangular offset']))) > 0))) = 0);
wr5 : (SIZEOF(QUERY(cor <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'CHARACTERIZED_OBJECT_RELATIONSHIP.RELATED_OBJECT') | (cor.name =
        'replicate base assignment') AND (
        'AUTOMOTIVE_DESIGN.RECTANGULAR_PATTERN' IN TYPEOF(cor.relating_object)
        ))) = 0) OR (SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | sa.description IN ['rectangular omit',
        'rectangular offset'])) > 0))) >= 1);
wr6 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | SIZEOF(QUERY(pdr <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)))) = 1)) = SIZEOF(USEDIN(pds,
        'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE'))))) = SIZEOF(QUERY(pd <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION')
         | 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)));
wr7 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.description <> 'circular omit') OR (
        SIZEOF(QUERY(pdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(pdr.items) = 1))) = 1))) = SIZEOF(
        USEDIN(pds,'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE'))))) = SIZEOF(
        QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)));
```

```
wr8 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
      SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | (sa.description <> 'circular omit') OR (
      SIZEOF(QUERY(pdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
      .used_representation)) AND (SIZEOF(QUERY(r_i <* pdr.
      used_representation.items | (
      'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (
      'AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN TYPEOF(r_i\
      measure_representation_item.value_component)) AND (r_i.name =
      'omit index'))) = 1))) = 1))) = SIZEOF(USEDIN(pds,
      'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE'))))) = SIZEOF(QUERY(pd <*
      USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION')
       | 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)));
wr9 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
      SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | NOT (sa.description IN ['circular offset',
      'rectangular omit']) OR (SIZEOF(QUERY(pdr <* USEDIN(sa,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.items) = 2))) = 1)))
      = SIZEOF(USEDIN(pds,'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE'))))) =
      SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') |
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)));
wr10 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') | (
       'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
       SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT.OF_SHAPE') | (sa.description <> 'circular offset') OR (
       SIZEOF(QUERY(pdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
       'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
       pdr.used_representation)) AND (SIZEOF(QUERY(r_i <* pdr.
       used_representation.items | (
       'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (
       'AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN TYPEOF(r_i\
       measure_representation_item.value_component)) AND (r_i.name =
       'offset index'))) = 1))) = 1))) = SIZEOF(USEDIN(pds,
       'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE'))))) = SIZEOF(QUERY(pd <*
       USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION')
        | 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)));
wr11 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') | (
       'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
       SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
       'SHAPE_ASPECT.OF_SHAPE') | (sa.description <> 'circular offset') OR (
```

```
        SIZEOF(QUERY(pdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
        pdr.used_representation)) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(r_i)) = 2
        ) AND (r_i.name = 'angular offset'))) = 1))) = 1))) = SIZEOF(USEDIN(
        pds,'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE'))))) = SIZEOF(QUERY(pd
        <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)));
wr12 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.description <> 'rectangular offsert')
        OR (SIZEOF(QUERY(pdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
        pdr.used_representation)) AND (SIZEOF(pdr.items) = 3))) = 1))) =
        SIZEOF(USEDIN(pds,'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE'))))) =
        SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)));
wr13 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | NOT (sa.description IN ['rectangular omit'
        ,'rectangular offset']) OR (SIZEOF(QUERY(pdr <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
        pdr.used_representation)) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | (
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (
        'AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN TYPEOF(r_i\
        measure_representation_item.value_component)) AND (r_i.name =
        'row index'))) = 1))) = 1))) = SIZEOF(USEDIN(pds,
        'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE'))))) = SIZEOF(QUERY(pd <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION')
         | 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)));
wr14 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | NOT (sa.description IN ['rectangular omit'
        ,'rectangular offset']) OR (SIZEOF(QUERY(pdr <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
```

```
        pdr.used_representation)) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | (
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (
        'AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN TYPEOF(r_i\
        measure_representation_item.value_component)) AND (r_i.name =
        'column index'))) = 1))) = 1))) = SIZEOF(USEDIN(pds,
        'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE'))))) = SIZEOF(QUERY(pd <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION')
         | 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)));
  wr15 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.description <> 'rectangular offset')
        OR (SIZEOF(QUERY(pdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
        pdr.used_representation)) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(r_i)) = 2) AND
         (r_i.name = 'offset distance'))) = 1))) = 1))) = SIZEOF(USEDIN(pds,
        'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE'))))) = SIZEOF(QUERY(pd <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION')
         | 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)));
  wr16 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.description <> 'rectangular offset')
        OR (SIZEOF(QUERY(pdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation)) AND (pdr.used_representation.description =
        'offset direction'))) = 1))) = SIZEOF(USEDIN(pds,
        'AUTOMOTIVE_DESIGN.SHAPE_ASPECT.OF_SHAPE'))))) = SIZEOF(QUERY(pd <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION')
         | 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)));
END_ENTITY;


ENTITY multi_language_attribute_assignment
 ABSTRACT SUPERTYPE;
 attribute_name : label;
 language : label;
 alternate_attribute_value : attribute_type;
 role : language_role;
END_ENTITY;


ENTITY named_unit
 SUPERTYPE OF (ONEOF(si_unit,conversion_based_unit,
               expression_conversion_based_unit,context_dependent_unit) ANDOR
```

```
                ONEOF(length_unit,mass_unit,time_unit,electric_current_unit,
                thermodynamic_temperature_unit,amount_of_substance_unit,
                luminous_intensity_unit,plane_angle_unit,solid_angle_unit,
                area_unit,volume_unit,ratio_unit));
 dimensions : dimensional_exponents;
END_ENTITY;


ENTITY next_assembly_usage_occurrence
 SUBTYPE OF (assembly_component_usage);
END_ENTITY;


ENTITY ngon_closed_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1;
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND ({3<=SIZEOF(pdr.used_representation.items)
        <=4}))) = 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | NOT (srwp_i.name IN ['orientation',
        'number of sides','circumscribed diameter','corner radius']))) > 0)))
        = 0;
  wr5 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(
        srwp_i)) AND (srwp_i.name = 'orientation'))) = 1))) = 1;
  wr6 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i)) AND
         ('AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN TYPEOF(srwp_i\
        measure_representation_item.value_component)) AND (srwp_i.name =
        'number of sides'))) = 1))) = 1;
  wr7 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
```

```
        used_representation.items | (srwp_i.name = 'circumscibed diameter')
        AND (SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1;
  wr8 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'corner radius'))) > 0))) =
         0) OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'corner radius') AND (
        SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1);
END_ENTITY;


ENTITY non_manifold_surface_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(it <* SELF.items | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM','AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D'
        ] * TYPEOF(it)) = 1))) = 0;
  wr2 : SIZEOF(QUERY(it <* SELF.items | SIZEOF([
        'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM'] * TYPEOF(it)) = 1)) > 0;
  wr3 : SIZEOF(QUERY(mi <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(it)) | NOT (
        'AUTOMOTIVE_DESIGN.NON_MANIFOLD_SURFACE_SHAPE_REPRESENTATION' IN
        TYPEOF(mi\mapped_item.mapping_source.mapped_representation)))) = 0;
  wr4 : SIZEOF(QUERY(fbsm <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
        SIZEOF(QUERY(cfs <* fbsm\face_based_surface_model.fbsm_faces | NOT (
        SIZEOF(QUERY(fa <* cfs.cfs_faces | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.FACE_SURFACE','AUTOMOTIVE_DESIGN.ORIENTED_FACE'] *
        TYPEOF(fa)) = 1))) = 0))) = 0))) = 0;
  wr5 : SIZEOF(QUERY(fbsm <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
        SIZEOF(QUERY(cfs <* fbsm\face_based_surface_model.fbsm_faces | NOT (
        SIZEOF(QUERY(f_sf <* QUERY(fa <* cfs.cfs_faces |
        'AUTOMOTIVE_DESIGN.FACE_SURFACE' IN TYPEOF(fa)) | NOT ((
        'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(f_sf)) OR (SIZEOF([
        'AUTOMOTIVE_DESIGN.OFFSET_SURFACE','AUTOMOTIVE_DESIGN.SURFACE_REPLICA'
        ] * TYPEOF(f_sf\face_surface.face_geometry)) = 1)))) = 0))) = 0))) = 0
        ;
  wr6 : SIZEOF(QUERY(fbsm <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
        SIZEOF(QUERY(cfs <* fbsm\face_based_surface_model.fbsm_faces | NOT (
        SIZEOF(QUERY(fa <* cfs.cfs_faces | NOT ((
```

```
        'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (
        nmsf_surface_check(fa\face_surface.face_geometry))))) = 0))) = 0))) =
        0;
wr7 : SIZEOF(QUERY(fbsm <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
        SIZEOF(QUERY(cfs <* fbsm\face_based_surface_model.fbsm_faces | NOT (
        SIZEOF(QUERY(fa <* cfs.cfs_faces | NOT ((
        'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(bnds
        <* fa.bounds | NOT (SIZEOF(['AUTOMOTIVE_DESIGN.EDGE_LOOP',
        'AUTOMOTIVE_DESIGN.VERTEX_LOOP'] * TYPEOF(bnds.bound)) = 1))) = 0))))
        = 0))) = 0))) = 0;
wr8 : SIZEOF(QUERY(fbsm <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
        SIZEOF(QUERY(cfs <* fbsm\face_based_surface_model.fbsm_faces | NOT (
        SIZEOF(QUERY(fa <* cfs.cfs_faces | NOT ((
        'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
        elp_fbnds <* QUERY(bnds <* fa.bounds | 'AUTOMOTIVE_DESIGN.EDGE_LOOP'
        IN TYPEOF(bnds)) | NOT (SIZEOF(QUERY(oe <* elp_fbnds.bound\path.
        edge_list | NOT ('AUTOMOTIVE_DESIGN.EDGE_CURVE' IN TYPEOF(oe.
        edge_element)))) = 0))) = 0)))) = 0))) = 0))) = 0;
wr9 : SIZEOF(QUERY(fbsm <* QUERY(it <* SELF.items |
        'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
        SIZEOF(QUERY(cfs <* fbsm\face_based_surface_model.fbsm_faces | NOT (
        SIZEOF(QUERY(fa <* cfs.cfs_faces | NOT ((
        'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
        elp_fbnds <* QUERY(bnds <* fa.bounds | 'AUTOMOTIVE_DESIGN.EDGE_LOOP'
        IN TYPEOF(bnds.bound)) | NOT (SIZEOF(QUERY(oe_cv <* QUERY(oe <*
        elp_fbnds.bound\path.edge_list | 'AUTOMOTIVE_DESIGN.EDGE_CURVE' IN
        TYPEOF(oe.edge_element)) | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.CURVE_REPLICA','AUTOMOTIVE_DESIGN.OFFSET_CURVE_3D',
        'AUTOMOTIVE_DESIGN.PCURVE'] * TYPEOF(oe_cv.edge_element\edge_curve.
        edge_geometry)) = 1))) = 0))) = 0)))) = 0))) = 0))) = 0;
wr10 : SIZEOF(QUERY(fbsm <* QUERY(it <* SELF.items |
         'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
         SIZEOF(QUERY(cfs <* fbsm\face_based_surface_model.fbsm_faces | NOT (
         SIZEOF(QUERY(fa <* cfs.cfs_faces | NOT ((
         'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
         elp_fbnds <* QUERY(bnds <* fa.bounds | 'AUTOMOTIVE_DESIGN.EDGE_LOOP'
         IN TYPEOF(bnds.bound)) | NOT (SIZEOF(QUERY(oe <* elp_fbnds.bound\path
         .edge_list | NOT (nmsf_curve_check(oe.edge_element\edge_curve.
         edge_geometry)))) = 0))) = 0)))) = 0))) = 0))) = 0;
wr11 : SIZEOF(QUERY(fbsm <* QUERY(it <* SELF.items |
         'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
         SIZEOF(QUERY(cfs <* fbsm\face_based_surface_model.fbsm_faces | NOT (
         SIZEOF(QUERY(fa <* cfs.cfs_faces | NOT ((
         'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
         elp_fbnds <* QUERY(bnds <* fa.bounds | 'AUTOMOTIVE_DESIGN.EDGE_LOOP'
         IN TYPEOF(bnds)) | NOT (SIZEOF(QUERY(oe <* elp_fbnds.bound\path.
         edge_list | NOT (('AUTOMOTIVE_DESIGN.VERTEX_POINT' IN TYPEOF(oe.
         edge_element.edge_start)) AND ('AUTOMOTIVE_DESIGN.VERTEX_POINT' IN
         TYPEOF(oe.edge_element.edge_end))))) = 0))) = 0)))) = 0))) = 0))) = 0
         ;
```

```
  wr12 : SIZEOF(QUERY(fbsm <* QUERY(it <* SELF.items |
         'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
         SIZEOF(QUERY(cfs <* fbsm\face_based_surface_model.fbsm_faces | NOT (
         SIZEOF(QUERY(fa <* cfs.cfs_faces | NOT ((
         'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
         elp_fbnds <* QUERY(bnds <* fa.bounds | 'AUTOMOTIVE_DESIGN.EDGE_LOOP'
         IN TYPEOF(bnds.bound)) | NOT (SIZEOF(QUERY(oe <* elp_fbnds.bound\path
         .edge_list | NOT ((SIZEOF(['AUTOMOTIVE_DESIGN.CARTESIAN_POINT',
         'AUTOMOTIVE_DESIGN.DEGENERATE_PCURVE',
         'AUTOMOTIVE_DESIGN.POINT_ON_CURVE',
         'AUTOMOTIVE_DESIGN.POINT_ON_SURFACE'] * TYPEOF(oe.edge_element.
         edge_start\vertex_point.vertex_geometry)) = 1) AND (SIZEOF([
         'AUTOMOTIVE_DESIGN.CARTESIAN_POINT',
         'AUTOMOTIVE_DESIGN.DEGENERATE_PCURVE',
         'AUTOMOTIVE_DESIGN.POINT_ON_CURVE',
         'AUTOMOTIVE_DESIGN.POINT_ON_SURFACE'] * TYPEOF(oe.edge_element.
         edge_start\vertex_point.vertex_geometry)) = 1)))) = 0))) = 0)))) = 0)
         )) = 0))) = 0;
  wr13 : SIZEOF(QUERY(fbsm <* QUERY(it <* SELF.items |
         'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
         SIZEOF(QUERY(cfs <* fbsm\face_based_surface_model.fbsm_faces | NOT (
         SIZEOF(QUERY(fa <* cfs.cfs_faces | NOT ((
         'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
         vlp_fbnds <* QUERY(bnds <* fa.bounds |
         'AUTOMOTIVE_DESIGN.VERTEX_LOOP' IN TYPEOF(bnds.bound)) | NOT (
         'AUTOMOTIVE_DESIGN.VERTEX_POINT' IN TYPEOF(vlp_fbnds.bound\
         vertex_loop.loop_vertex)))) = 0)))) = 0))) = 0))) = 0;
  wr14 : SIZEOF(QUERY(fbsm <* QUERY(it <* SELF.items |
         'AUTOMOTIVE_DESIGN.FACE_BASED_SURFACE_MODEL' IN TYPEOF(it)) | NOT (
         SIZEOF(QUERY(cfs <* fbsm\face_based_surface_model.fbsm_faces | NOT (
         SIZEOF(QUERY(fa <* cfs.cfs_faces | NOT ((
         'AUTOMOTIVE_DESIGN.ADVANCED_FACE' IN TYPEOF(fa)) OR (SIZEOF(QUERY(
         vlp_fbnds <* QUERY(bnds <* fa.bounds |
         'AUTOMOTIVE_DESIGN.VERTEX_LOOP' IN TYPEOF(bnds.bound)) | NOT (SIZEOF(
         ['AUTOMOTIVE_DESIGN.CARTESIAN_POINT',
         'AUTOMOTIVE_DESIGN.DEGENERATE_PCURVE',
         'AUTOMOTIVE_DESIGN.POINT_ON_CURVE',
         'AUTOMOTIVE_DESIGN.POINT_ON_SURFACE'] * TYPEOF(vlp_fbnds.bound\
         vertex_loop.loop_vertex\vertex_point.vertex_geometry)) = 1))) = 0))))
          = 0))) = 0))) = 0;
END_ENTITY;


ENTITY occurrence_usage_definition
 SUBTYPE OF (configuration_design);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_OCCURRENCE' IN TYPEOF(SELF.
        design);
  wr2 : ('AUTOMOTIVE_DESIGN.CONFIGURABLE_ITEM' IN TYPEOF(SELF.configuration))
        AND (SELF.configuration.description = 'product specification');
END_ENTITY;
```

```
ENTITY offset_curve_2d
 SUBTYPE OF (curve);
 basis_curve : curve;
 distance : length_measure;
 self_intersect : LOGICAL;
 WHERE
  wr1 : basis_curve.dim = 2;
END_ENTITY;


ENTITY offset_curve_3d
 SUBTYPE OF (curve);
 basis_curve : curve;
 distance : length_measure;
 self_intersect : LOGICAL;
 ref_direction : direction;
 WHERE
  wr1 : (basis_curve.dim = 3) AND (ref_direction.dim = 3);
END_ENTITY;


ENTITY offset_surface
 SUBTYPE OF (surface);
 basis_surface : surface;
 distance : length_measure;
 self_intersect : LOGICAL;
END_ENTITY;


ENTITY one_direction_repeat_factor
 SUBTYPE OF (geometric_representation_item);
 repeat_factor : vector;
END_ENTITY;


ENTITY open_path_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) <= 1;
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(pdr.used_representation.items) = 1)
        )) <= 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'orientation') AND (
```

```
        'AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i)))) = 1))) = 1;
  wr5 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PATH_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation)))) = 1;
END_ENTITY;


ENTITY open_shell
 SUBTYPE OF (connected_face_set);
END_ENTITY;


ENTITY ordinate_dimension
 SUBTYPE OF (projection_directed_callout);
END_ENTITY;


ENTITY organization;
 id : OPTIONAL identifier;
 name : label;
 description : text;
END_ENTITY;


ENTITY organization_assignment
 ABSTRACT SUPERTYPE;
 assigned_organization : organization;
 role : organization_role;
END_ENTITY;


ENTITY organization_relationship;
 name : label;
 description : text;
 relating_organization : organization;
 related_organization : organization;
END_ENTITY;


ENTITY organization_role;
 name : label;
 description : text;
END_ENTITY;


ENTITY organizational_address
 SUBTYPE OF (address);
 organizations : SET[1:?] OF organization;
 description : text;
END_ENTITY;


ENTITY oriented_closed_shell
 SUBTYPE OF (closed_shell);
```

```
 closed_shell_element : closed_shell;
 orientation : BOOLEAN;
 DERIVE
  SELF\connected_face_set.cfs_faces : SET[1:?] OF face := conditional_reverse(
                                      SELF.orientation,SELF.
                                      closed_shell_element.cfs_faces);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.ORIENTED_CLOSED_SHELL' IN TYPEOF(SELF.
        closed_shell_element));
END_ENTITY;


ENTITY oriented_edge
 SUBTYPE OF (edge);
 edge_element : edge;
 orientation : BOOLEAN;
 DERIVE
  SELF\edge.edge_start : vertex := boolean_choose(SELF.orientation,SELF.
                         edge_element.edge_start,SELF.edge_element.edge_end);
  SELF\edge.edge_end : vertex := boolean_choose(SELF.orientation,SELF.
                       edge_element.edge_end,SELF.edge_element.edge_start);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.ORIENTED_EDGE' IN TYPEOF(SELF.edge_element));
END_ENTITY;


ENTITY oriented_face
 SUBTYPE OF (face);
 face_element : face;
 orientation : BOOLEAN;
 DERIVE
  SELF\face.bounds : SET[1:?] OF face_bound := conditional_reverse(SELF.
                     orientation,SELF.face_element.bounds);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.ORIENTED_FACE' IN TYPEOF(SELF.face_element));
END_ENTITY;


ENTITY oriented_open_shell
 SUBTYPE OF (open_shell);
 open_shell_element : open_shell;
 orientation : BOOLEAN;
 DERIVE
  SELF\connected_face_set.cfs_faces : SET[1:?] OF face := conditional_reverse(
                                      SELF.orientation,SELF.open_shell_element
                                      .cfs_faces);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.ORIENTED_OPEN_SHELL' IN TYPEOF(SELF.
        open_shell_element));
END_ENTITY;


ENTITY oriented_path
 SUBTYPE OF (path);
```

```
 path_element : path;
 orientation : BOOLEAN;
 DERIVE
  SELF\path.edge_list : LIST[1:?] OF UNIQUE oriented_edge :=
                        conditional_reverse(SELF.orientation,SELF.path_element
                        .edge_list);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.ORIENTED_PATH' IN TYPEOF(SELF.path_element));
END_ENTITY;


ENTITY outer_boundary_curve
 SUBTYPE OF (boundary_curve);
END_ENTITY;


ENTITY over_riding_styled_item
 SUBTYPE OF (styled_item);
 over_ridden_style : styled_item;
END_ENTITY;


ENTITY package_product_concept_feature
 SUBTYPE OF (product_concept_feature);
END_ENTITY;


ENTITY pair_actuator;
 actuated_pair : kinematic_pair;
 name : label;
 UNIQUE
  ur1 : actuated_pair;
END_ENTITY;


ENTITY pair_value;
 applies_to_pair : kinematic_pair;
END_ENTITY;


ENTITY parabola
 SUBTYPE OF (conic);
 focal_dist : length_measure;
 WHERE
  wr1 : focal_dist <> 0.0;
END_ENTITY;


ENTITY parallel_offset
 SUBTYPE OF (derived_shape_aspect);
 offset : measure_with_unit;
 WHERE
  wr1 : SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
END_ENTITY;
```

```
ENTITY parallelism_tolerance
 SUBTYPE OF (geometric_tolerance_with_datum_reference);
 WHERE
  wr1 : SIZEOF(SELF\geometric_tolerance_with_datum_reference.datum_system) = 1
        ;
END_ENTITY;


ENTITY parametric_representation_context
 SUBTYPE OF (representation_context);
END_ENTITY;


ENTITY partial_circular_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1;
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(pdr.used_representation.items) = 3)
        )) <= 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'orientation') AND (
        'AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i)))) = 1))) = 1;
  wr5 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'radius') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1;
  wr6 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'sweep angle') AND (SIZEOF(
        ['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) =
         2))) = 1))) = 1;
END_ENTITY;


ENTITY path
```

```
 SUPERTYPE OF (ONEOF(edge_loop,oriented_path))
 SUBTYPE OF (topological_representation_item);
 edge_list : LIST[1:?] OF UNIQUE oriented_edge;
 WHERE
  wr1 : path_head_to_tail(SELF);
END_ENTITY;


ENTITY path_feature_component
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SELF.description IN ['partial circular','complete circular','linar',
        'complex'];
  wr3 : (SELF.description <> 'complex') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)))) = 0);
  wr4 : NOT (SELF.description IN ['partial circular','complete circular',
        'linar']) OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'orientation') AND (
        'AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i)))) = 1))) = 1);
  wr5 : (SELF.description <> 'partial circular') OR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(pdr.used_representation.items) = 3)
        )) = 1);
  wr6 : (SELF.description <> 'partial circular') OR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'radius') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1);
  wr7 : (SELF.description <> 'partial circular') OR (SIZEOF(QUERY(pdr <* USEDIN
        (SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'sweep angle') AND (SIZEOF(
        ['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) =
         2))) = 1))) = 1);
  wr8 : (SELF.description <> 'complete circular') OR (SIZEOF(QUERY(pdr <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
```

```
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(pdr.used_representation.items) = 2)
        )) = 1);
  wr9 : (SELF.description <> 'complete circular') OR (SIZEOF(QUERY(pdr <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'radius') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1);
  wr10 : (SELF.description <> 'linear') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
         'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
         'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
         pdr.used_representation)) AND (SIZEOF(pdr.used_representation.items)
         = 2))) = 1);
  wr11 : (SELF.description <> 'linear') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
         'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
         'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
         pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
         used_representation.items | (srwp_i.name = 'distance') AND (SIZEOF([
         'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2))
         ) = 1))) = 1);
  wr12 : (SELF.description <> 'complex') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
         'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
         'AUTOMOTIVE_DESIGN.PATH_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
         used_representation)) AND (pdr.used_representation.description =
         'path geometry') AND (SIZEOF(QUERY(srwp_i <* pdr.used_representation.
         items | (srwp_i.name = 'path curve') AND ('AUTOMOTIVE_DESIGN.CURVE'
         IN TYPEOF(srwp_i)))) = 1))) = 1);
END_ENTITY;


ENTITY path_shape_representation
 SUBTYPE OF (representation);
 WHERE
  wr1 : SIZEOF(SELF.items) = 1;
  wr2 : SIZEOF(QUERY(i <* SELF.items | 'AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF(i))
        ) = 1;
END_ENTITY;


ENTITY pcurve
 SUBTYPE OF (curve);
 basis_surface : surface;
 reference_to_curve : definitional_representation;
 WHERE
  wr1 : SIZEOF(reference_to_curve\representation.items) = 1;
```

```
  wr2 : 'AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF(reference_to_curve\representation.
        items[1]);
  wr3 : reference_to_curve\representation.items[1]\
        geometric_representation_item.dim = 2;
END_ENTITY;


ENTITY perpendicular_to
 SUBTYPE OF (derived_shape_aspect);
 WHERE
  wr1 : SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
END_ENTITY;


ENTITY perpendicularity_tolerance
 SUBTYPE OF (geometric_tolerance_with_datum_reference);
 WHERE
  wr1 : SIZEOF(SELF\geometric_tolerance_with_datum_reference.datum_system) = 1
        ;
END_ENTITY;


ENTITY person;
 id : identifier;
 last_name : OPTIONAL label;
 first_name : OPTIONAL label;
 middle_names : OPTIONAL LIST[1:?] OF label;
 prefix_titles : OPTIONAL LIST[1:?] OF label;
 suffix_titles : OPTIONAL LIST[1:?] OF label;
 WHERE
  wr1 : EXISTS(last_name) OR EXISTS(first_name);
END_ENTITY;


ENTITY person_and_organization;
 name : label;
 description : text;
 the_person : person;
 the_organization : organization;
END_ENTITY;


ENTITY person_and_organization_assignment
 ABSTRACT SUPERTYPE;
 assigned_person_and_organization : person_and_organization;
 role : person_and_organization_role;
END_ENTITY;


ENTITY person_and_organization_role;
 name : label;
 description : text;
END_ENTITY;
```

```
ENTITY person_assignment
 ABSTRACT SUPERTYPE;
 assigned_person : person;
 role : person_role;
END_ENTITY;


ENTITY person_role;
 name : label;
 description : text;
END_ENTITY;


ENTITY personal_address
 SUBTYPE OF (address);
 people : SET[1:?] OF person;
 description : text;
END_ENTITY;


ENTITY physically_modelled_product_definition
 SUBTYPE OF (product_definition_with_associated_documents);
 WHERE
  wr1 : SELF.frame_of_reference\application_context_element.name =
        'physical occurrence';
  wr2 : (SIZEOF(SELF.documentation_ids) = 1) AND (SIZEOF(QUERY(d <* SELF.
        documentation_ids | d.kind.product_data_type = 'physical model')) = 1)
        ;
  wr3 : SIZEOF(QUERY(aoa <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | aoa.role.name = 'location')
        ) = 1;
  wr4 : (SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd))) = 0) OR (
        SIZEOF(QUERY(pds <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1) AND (SIZEOF(
        QUERY(pdr <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINITION_REPRESENTATION' IN TYPEOF(pdr))
        AND ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation)) AND (SIZEOF(pdr.used_representation.items) = 1))
        ) = 1))) = 1);
END_ENTITY;


ENTITY placed_datum_target_feature
 SUBTYPE OF (datum_target);
 WHERE
  wr1 : SELF.description IN ['point','line','rectangle','circle'];
  wr2 : SIZEOF(QUERY(pd <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.' + 'DEFINITION') | NOT (SIZEOF
```

```
      (QUERY(pdr <* USEDIN(pd,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'AUTOMOTIVE_DESIGN.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation))) = 1))) = 0;
wr3 : SIZEOF(QUERY(pd <* USEDIN(SELF,
      'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.' + 'DEFINITION') | NOT (SIZEOF
      (QUERY(impl_rep <* QUERY(pdr <* USEDIN(pd,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'AUTOMOTIVE_DESIGN.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) | NOT (SIZEOF(QUERY(it <* impl_rep.
      used_representation.items | ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(
      it)) AND (it.name = 'orientation'))) = 1))) = 0))) = 0;
wr4 : (NOT (SELF.description = 'point')) OR (SIZEOF(QUERY(pd <* USEDIN(SELF,
      'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.' + 'DEFINITION') | NOT (SIZEOF
      (QUERY(impl_rep <* QUERY(pdr <* USEDIN(pd,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'AUTOMOTIVE_DESIGN.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) | NOT (SIZEOF(impl_rep.
      used_representation.items) = 1))) = 0))) = 0);
wr5 : (NOT (SELF.description IN ['line','circle'])) OR (SIZEOF(QUERY(pd <*
      USEDIN(SELF,'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.' + 'DEFINITION')
       | NOT (SIZEOF(QUERY(impl_rep <* QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | 'AUTOMOTIVE_DESIGN.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) | NOT (SIZEOF(impl_rep.
      used_representation.items) = 2))) = 0))) = 0);
wr6 : (NOT (SELF.description = 'rectangle')) OR (SIZEOF(QUERY(pd <* USEDIN(
      SELF,'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.' + 'DEFINITION') | NOT (
      SIZEOF(QUERY(impl_rep <* QUERY(pdr <* USEDIN(pd,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'AUTOMOTIVE_DESIGN.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) | NOT (SIZEOF(impl_rep.
      used_representation.items) = 3))) = 0))) = 0);
wr7 : (NOT (SELF.description = 'circle')) OR (SIZEOF(QUERY(pd <* USEDIN(SELF
      ,'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.DEFINITION') | NOT (SIZEOF(
      QUERY(impl_rep <* QUERY(pdr <* USEDIN(pd,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'AUTOMOTIVE_DESIGN.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) | NOT (SIZEOF(QUERY(it <* impl_rep.
      used_representation.items | (SIZEOF([
      'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (
      it.name = 'target diameter'))) = 1))) = 0))) = 0);
wr8 : (NOT (SELF.description = 'line')) OR (SIZEOF(QUERY(pd <* USEDIN(SELF,
      'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.DEFINITION') | NOT (SIZEOF(
      QUERY(impl_rep <* QUERY(pdr <* USEDIN(pd,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      'AUTOMOTIVE_DESIGN.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) | NOT (SIZEOF(QUERY(it <* impl_rep.
      used_representation.items | (SIZEOF([
      'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (
```

```
        it.name = 'target length'))) = 1))) = 0))) = 0);
  wr9 : (NOT (SELF.description = 'rectangle')) OR (SIZEOF(QUERY(pd <* USEDIN(
        SELF,'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.DEFINITION') | NOT (SIZEOF
        (QUERY(impl_rep <* QUERY(pdr <* USEDIN(pd,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) | NOT (SIZEOF(QUERY(it <* impl_rep.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND (
        it.name = 'target length'))) = 1))) = 0))) = 0);
  wr10 : (NOT (SELF.description = 'rectangle')) OR (SIZEOF(QUERY(pd <* USEDIN(
         SELF,'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION.DEFINITION') | NOT (
         SIZEOF(QUERY(impl_rep <* QUERY(pdr <* USEDIN(pd,'AUTOMOTIVE_DESIGN.'
         + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
         'AUTOMOTIVE_DESIGN.' + 'SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
         TYPEOF(pdr.used_representation)) | NOT (SIZEOF(QUERY(it <* impl_rep.
         used_representation.items | (SIZEOF([
         'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
         (it.name = 'target width'))) = 1))) = 0))) = 0);
END_ENTITY;


ENTITY placed_feature
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : SIZEOF(['AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION',
        'AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_SOLID',
        'AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_PANEL'] * TYPEOF(SELF.of_shape.
        definition)) = 1;
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.name = 'feature definition placement') AND (SIZEOF
        (QUERY(r_i <* pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(r_i)) AND (SIZEOF(QUERY(pdr
        <* USEDIN(r_i\mapped_item.mapping_source.mapped_representation,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.' +
        'USED_REPRESENTATION') | 'AUTOMOTIVE_DESIGN.FEATURE_DEFINITION' IN
        TYPEOF(pdr.definition\property_definition.definition))) = 1))) = 1)))
        = 1;
END_ENTITY;


ENTITY placement
 SUPERTYPE OF (ONEOF(axis1_placement,axis2_placement_2d,axis2_placement_3d))
 SUBTYPE OF (geometric_representation_item);
 location : cartesian_point;
END_ENTITY;


ENTITY planar_box
 SUBTYPE OF (planar_extent);
```

```
 placement : axis2_placement;
END_ENTITY;


ENTITY planar_curve_pair
 SUBTYPE OF (kinematic_pair);
 curve_1 : curve;
 curve_2 : curve;
 orientation : BOOLEAN;
 WHERE
  wr1 : frame_associated_to_background(SELF\kinematic_pair.
        pair_placement_in_first_link_context,curve_1);
  wr2 : frame_associated_to_background(SELF\kinematic_pair.
        pair_placement_in_second_link_context,curve_2);
END_ENTITY;


ENTITY planar_curve_pair_range
 SUBTYPE OF (simple_pair_range);
 SELF\simple_pair_range.applies_to_pair : planar_curve_pair;
 range_on_curve_1 : trimmed_curve;
 range_on_curve_2 : trimmed_curve;
 WHERE
  wr1 : SELF\simple_pair_range.applies_to_pair\planar_curve_pair.curve_1 :=:
        range_on_curve_1.basis_curve;
  wr2 : SELF\simple_pair_range.applies_to_pair\planar_curve_pair.curve_2 :=:
        range_on_curve_2.basis_curve;
END_ENTITY;


ENTITY planar_extent
 SUBTYPE OF (geometric_representation_item);
 size_in_x : length_measure;
 size_in_y : length_measure;
END_ENTITY;


ENTITY planar_pair
 SUBTYPE OF (kinematic_pair);
END_ENTITY;


ENTITY planar_pair_range
 SUBTYPE OF (simple_pair_range);
 SELF\simple_pair_range.applies_to_pair : planar_pair;
 lower_limit_actual_rotation : rotational_range_measure;
 upper_limit_actual_rotation : rotational_range_measure;
 lower_limit_actual_translation_x : translational_range_measure;
 upper_limit_actual_translation_x : translational_range_measure;
 lower_limit_actual_translation_y : translational_range_measure;
 upper_limit_actual_translation_y : translational_range_measure;
 WHERE
  wr1 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        lower_limit_actual_rotation)) OR ('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE'
```

```
        IN TYPEOF(upper_limit_actual_rotation))) XOR (
        lower_limit_actual_rotation < upper_limit_actual_rotation);
  wr2 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        lower_limit_actual_translation_x)) OR (
        'AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        upper_limit_actual_translation_x))) XOR (
        lower_limit_actual_translation_x < upper_limit_actual_translation_x);
  wr3 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        lower_limit_actual_translation_y)) OR (
        'AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        upper_limit_actual_translation_y))) XOR (
        lower_limit_actual_translation_y < upper_limit_actual_translation_y);
END_ENTITY;


ENTITY planar_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : planar_pair;
 actual_rotation : plane_angle_measure;
 actual_translation_x : length_measure;
 actual_translation_y : length_measure;
END_ENTITY;


ENTITY plane
 SUBTYPE OF (elementary_surface);
END_ENTITY;


ENTITY plane_angle_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.PLANE_ANGLE_UNIT' IN TYPEOF(SELF\measure_with_unit.
        unit_component);
END_ENTITY;


ENTITY plane_angle_unit
 SUBTYPE OF (named_unit);
 WHERE
  wr1 : (SELF\named_unit.dimensions.length_exponent = 0.0) AND (SELF\
        named_unit.dimensions.mass_exponent = 0.0) AND (SELF\named_unit.
        dimensions.time_exponent = 0.0) AND (SELF\named_unit.dimensions.
        electric_current_exponent = 0.0) AND (SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 0.0) AND (SELF\named_unit.
        dimensions.amount_of_substance_exponent = 0.0) AND (SELF\named_unit.
        dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;


ENTITY plus_minus_tolerance;
 range : tolerance_method_definition;
 toleranced_dimension : dimensional_characteristic;
 UNIQUE
```

```
  ur1 : toleranced_dimension;
END_ENTITY;


ENTITY pocket
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SELF.description IN ['open rectangular','closed rectangular',
        'non-rectangular'];
  wr2 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'sweep path') AND (sa.
        description = 'path feature component occurrence') AND (SIZEOF(QUERY(
        sdr <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sdr)) AND (
        sdr.name = 'path feature component usage') AND (
        'AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN TYPEOF(sdr.
        relating_shape_aspect)) AND (sdr.relating_shape_aspect.description =
        'linear'))) = 1))) = 1))) = 1;
  wr3 : NOT (SELF.description = 'non-rectangular') OR (SIZEOF(QUERY(pds <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION')
         | ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'boundary') AND (sa.description
        = 'profile occurrence') AND (SIZEOF(QUERY(sdr <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sdr)) AND (
        sdr.name = 'profile usage') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.NGON_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.CLOSED_PATH_PROFILE',
        'AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE',
        'AUTOMOTIVE_DESIGN.PARTIAL_CIRCULAR_PROFILE',
        'AUTOMOTIVE_DESIGN.ROUNDED_U_PROFILE','AUTOMOTIVE_DESIGN.VEE_PROFILE',
        'AUTOMOTIVE_DESIGN.TEE_PROFILE','AUTOMOTIVE_DESIGN.OPEN_PATH_PROFILE']
         * TYPEOF(sdr.relating_shape_aspect)) = 1))) = 1))) = 1))) = 1);
  wr4 : NOT (SELF.description = 'closed rectangular') OR (SIZEOF(QUERY(pds <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION')
         | ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'boundary') AND (sa.description
        = 'closed profile occurrence') AND (SIZEOF(QUERY(sdr <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sdr)) AND (
        sdr.name = 'closed profile usage') AND (
        'AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE' IN TYPEOF(sdr.
        relating_shape_aspect)))) = 1))) = 1))) = 1);
```

```
  wr5 : NOT (SELF.description = 'open rectangular') OR (SIZEOF(QUERY(pds <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION')
         | ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'boundary') AND (sa.description
        = 'closed profile occurrence') AND (SIZEOF(QUERY(sdr <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sdr)) AND (
        sdr.name = 'closed profile usage') AND (
        'AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE' IN TYPEOF(sdr.
        relating_shape_aspect)))) = 1))) = 1))) = 1);
  wr6 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'bottom condition') AND (sa.
        description = 'pocket bottom occurrence') AND (SIZEOF(QUERY(fcr <*
        USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr)) AND
         (fcr.name = 'pocket bottom usage') AND (
        'AUTOMOTIVE_DESIGN.POCKET_BOTTOM' IN TYPEOF(fcr.relating_shape_aspect)
        ))) = 1))) = 1))) = 1;
  wr7 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'taper') AND (sa.description =
        'taper occurrence') AND (SIZEOF(QUERY(fcr <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr)) AND
         (fcr.name = 'taper usage') AND ('AUTOMOTIVE_DESIGN.TAPER' IN TYPEOF(
        fcr.relating_shape_aspect)))) = 1))) = 1))) <= 1;
END_ENTITY;


ENTITY pocket_bottom
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SELF.description IN ['planar','non-planar','through'];
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(pdr.used_representation.items) = 1)
         AND (SIZEOF(QUERY(srwp_i <* pdr.used_representation.items | (
```

```
      'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i))
       AND (srwp_i.name = 'bottom location') AND (srwp_i.description IN [
      'path start','path end']))) = 1))) = 1;
wr5 : (SELF.description <> 'non-planar') OR ((SIZEOF(QUERY(pdr <* USEDIN(
      SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF([
      'AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION',
      'AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION'] * TYPEOF(pdr.
      used_representation)) = 1))) = 1) AND (SIZEOF(QUERY(pdr <* USEDIN(SELF
      ,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (SIZEOF([
      'AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION',
      'AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION'] * TYPEOF(pdr.
      used_representation)) = 1) AND (pdr.used_representation.description =
      'floor face'))) = 1));
wr6 : (SELF.description <> 'planar') OR ((SIZEOF(QUERY(pdr <* USEDIN(SELF,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.LOCATION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
      used_representation)))) = 1) AND (SIZEOF(QUERY(pdr <* USEDIN(SELF,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.LOCATION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
      used_representation)) AND (pdr.used_representation.description =
      'floor location'))) = 1));
wr7 : (SELF.description <> 'planar') OR ((SIZEOF(QUERY(pdr <* USEDIN(SELF,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
      used_representation)))) = 1) AND (SIZEOF(QUERY(pdr <* USEDIN(SELF,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
      used_representation)) AND (pdr.used_representation.description =
      'floor normal'))) = 1));
wr8 : (SELF.description <> 'through') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.used_representation.
      items) = 1))) = 1);
wr9 : NOT (SELF.description IN ['planar','non-planar']) OR (SIZEOF(QUERY(pdr
      <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
      .used_representation)) AND ({1<=SIZEOF(pdr.used_representation.items)
      <=2}))) = 1);
wr10 : NOT (SELF.description IN ['planar','non-planar']) OR (SIZEOF(QUERY(
       pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
       'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
       pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
       used_representation.items | (SIZEOF([
       'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
       AND (srwp_i.name = 'floor radius'))) = 1))) <= 1);
wr11 : (SIZEOF(QUERY(sar <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
```

```
         'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') |
         'AUTOMOTIVE_DESIGN.FAETURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar))) =
          1) AND (SIZEOF(QUERY(fcr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
         'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (
         'AUTOMOTIVE_DESIGN.FAETURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr))
         AND (fcr.name = 'pocket bottom usage') AND (
         'AUTOMOTIVE_DESIGN.POCKET' IN TYPEOF(fcr.related_shape_aspect.
         of_shape.definition)) AND (fcr.related_shape_aspect.of_shape.
         definition.name = 'bottom condition') AND (fcr.related_shape_aspect.
         of_shape.definition.description = 'pocket bottom occurrence'))) = 1);
END_ENTITY;


ENTITY point
 SUPERTYPE OF (ONEOF(cartesian_point,point_on_curve,point_on_surface,
               point_replica,degenerate_pcurve))
 SUBTYPE OF (geometric_representation_item);
END_ENTITY;


ENTITY point_on_curve
 SUBTYPE OF (point);
 basis_curve : curve;
 point_parameter : parameter_value;
END_ENTITY;


ENTITY point_on_planar_curve_pair
 SUBTYPE OF (kinematic_pair);
 pair_curve : curve;
 orientation : BOOLEAN;
 WHERE
  wr1 : frame_associated_to_background(SELF\kinematic_pair.
        pair_placement_in_first_link_context,pair_curve);
END_ENTITY;


ENTITY point_on_planar_curve_pair_range
 SUBTYPE OF (simple_pair_range);
 SELF\simple_pair_range.applies_to_pair : point_on_planar_curve_pair;
 range_on_pair_curve : trimmed_curve;
 lower_limit_yaw : rotational_range_measure;
 upper_limit_yaw : rotational_range_measure;
 lower_limit_pitch : rotational_range_measure;
 upper_limit_pitch : rotational_range_measure;
 lower_limit_roll : rotational_range_measure;
 upper_limit_roll : rotational_range_measure;
 WHERE
  wr1 : SELF\simple_pair_range.applies_to_pair\point_on_planar_curve_pair.
        pair_curve :=: range_on_pair_curve.basis_curve;
  wr2 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(lower_limit_yaw)) OR (
        'AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(upper_limit_yaw))) XOR (
        lower_limit_yaw < upper_limit_yaw);
  wr3 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(lower_limit_pitch)) OR
```

```
        ('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(upper_limit_pitch)))
       XOR (lower_limit_pitch < upper_limit_pitch);
  wr4 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(lower_limit_roll)) OR
       ('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(upper_limit_roll))) XOR
        (lower_limit_roll < upper_limit_roll);
END_ENTITY;


ENTITY point_on_planar_curve_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : point_on_planar_curve_pair;
 actual_point_on_curve : point_on_curve;
 input_orientation : spatial_rotation;
 DERIVE
  actual_orientation : ypr_rotation := convert_spatial_to_ypr_rotation(SELF\
                       pair_value.applies_to_pair,input_orientation);
 WHERE
  wr1 : SELF\pair_value.applies_to_pair\point_on_planar_curve_pair.pair_curve
       :=: actual_point_on_curve.basis_curve;
END_ENTITY;


ENTITY point_on_surface
 SUBTYPE OF (point);
 basis_surface : surface;
 point_parameter_u : parameter_value;
 point_parameter_v : parameter_value;
END_ENTITY;


ENTITY point_on_surface_pair
 SUBTYPE OF (kinematic_pair);
 pair_surface : surface;
 WHERE
  wr1 : frame_associated_to_background(SELF\kinematic_pair.
       pair_placement_in_first_link_context,pair_surface);
END_ENTITY;


ENTITY point_on_surface_pair_range
 SUBTYPE OF (simple_pair_range);
 SELF\simple_pair_range.applies_to_pair : point_on_surface_pair;
 range_on_pair_surface : rectangular_trimmed_surface;
 lower_limit_yaw : rotational_range_measure;
 upper_limit_yaw : rotational_range_measure;
 lower_limit_pitch : rotational_range_measure;
 upper_limit_pitch : rotational_range_measure;
 lower_limit_roll : rotational_range_measure;
 upper_limit_roll : rotational_range_measure;
 WHERE
  wr1 : SELF\simple_pair_range.applies_to_pair\point_on_surface_pair.
       pair_surface :=: range_on_pair_surface.basis_surface;
  wr2 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(lower_limit_yaw)) OR (
       'AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(upper_limit_yaw))) XOR (
```

```
        lower_limit_yaw < upper_limit_yaw);
  wr3 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(lower_limit_pitch)) OR
         ('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(upper_limit_pitch)))
        XOR (lower_limit_pitch < upper_limit_pitch);
  wr4 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(lower_limit_roll)) OR
        ('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(upper_limit_roll))) XOR
         (lower_limit_roll < upper_limit_roll);
END_ENTITY;


ENTITY point_on_surface_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : point_on_surface_pair;
 actual_point_on_surface : point_on_surface;
 input_orientation : spatial_rotation;
 DERIVE
  actual_orientation : ypr_rotation := convert_spatial_to_ypr_rotation(SELF\
                       pair_value.applies_to_pair,input_orientation);
 WHERE
  wr1 : SELF\pair_value.applies_to_pair\point_on_surface_pair.pair_surface :=:
         actual_point_on_surface.basis_surface;
END_ENTITY;


ENTITY point_placement_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(i <* SELF.items | SIZEOF([
        'AUTOMOTIVE_DESIGN.COMPOUND_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM','AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT_3D'
        ] * TYPEOF(i)) <> 1)) = 0;
  wr2 : SIZEOF(QUERY(i <* SELF.items | SIZEOF([
        'AUTOMOTIVE_DESIGN.COMPOUND_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.MAPPED_ITEM'] * TYPEOF(i)) = 1)) > 0;
END_ENTITY;


ENTITY point_replica
 SUBTYPE OF (point);
 parent_pt : point;
 transformation : cartesian_transformation_operator;
 WHERE
  wr1 : transformation.dim = parent_pt.dim;
  wr2 : acyclic_point_replica(SELF,parent_pt);
END_ENTITY;


ENTITY point_style;
 name : label;
 marker : marker_select;
 marker_size : size_select;
 marker_colour : colour;
END_ENTITY;
```

```
ENTITY poly_loop
 SUBTYPE OF (loop,geometric_representation_item);
 polygon : LIST[3:?] OF UNIQUE cartesian_point;
END_ENTITY;


ENTITY polyline
 SUBTYPE OF (bounded_curve);
 points : LIST[2:?] OF cartesian_point;
END_ENTITY;


ENTITY position_tolerance
 SUBTYPE OF (geometric_tolerance_with_datum_reference);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT' IN
         TYPEOF(SELF));
  wr2 : SIZEOF(SELF\geometric_tolerance_with_datum_reference.datum_system) <=
        3;
END_ENTITY;


ENTITY pre_defined_colour
 SUBTYPE OF (pre_defined_item,colour);
END_ENTITY;


ENTITY pre_defined_curve_font
 SUBTYPE OF (pre_defined_item);
END_ENTITY;


ENTITY pre_defined_item;
 name : label;
END_ENTITY;


ENTITY pre_defined_marker
 SUBTYPE OF (pre_defined_item);
END_ENTITY;


ENTITY pre_defined_point_marker_symbol
 SUBTYPE OF (pre_defined_marker);
 WHERE
  wr1 : SELF.name IN ['asterisk','circle','dot','plus','square','triangle','x'
        ];
END_ENTITY;


ENTITY pre_defined_symbol
 SUBTYPE OF (pre_defined_item);
END_ENTITY;
```

```
ENTITY pre_defined_terminator_symbol
 SUBTYPE OF (pre_defined_symbol);
 WHERE
  wr1 : SELF.name IN ['blanked arrow','blanked box','blanked dot',
        'dimension origin','filled arrow','filled box','filled dot',
        'integral symbol','open arrow','slash','unfilled arrow'];
END_ENTITY;


ENTITY pre_defined_text_font
 SUBTYPE OF (pre_defined_item);
END_ENTITY;


ENTITY precision_qualifier;
 precision_value : INTEGER;
END_ENTITY;


ENTITY presentation_area
 SUBTYPE OF (presentation_representation);
 WHERE
  wr1 : ((SIZEOF(QUERY(ais <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'AREA_IN_SET.AREA') | SIZEOF(USEDIN(ais,'AUTOMOTIVE_DESIGN.' +
        'PRESENTATION_SIZE.UNIT')) = 1)) > 0) OR (SIZEOF(USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PRESENTATION_SIZE.UNIT')) = 1));
END_ENTITY;


ENTITY presentation_layer_assignment;
 name : label;
 description : text;
 assigned_items : SET[1:?] OF layered_item;
END_ENTITY;


ENTITY presentation_layer_usage;
 assignment : presentation_layer_assignment;
 presentation : presentation_representation;
 UNIQUE
  ur1 : assignment,presentation;
END_ENTITY;


ENTITY presentation_representation
 SUBTYPE OF (representation);
 WHERE
  wr1 : SELF\representation.context_of_items\geometric_representation_context.
        coordinate_space_dimension = 2;
  wr2 : 'AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_CONTEXT' IN TYPEOF(SELF\
        representation.context_of_items);
END_ENTITY;
```

```
ENTITY presentation_set;
 INVERSE
  areas : SET[1:?] OF area_in_set FOR in_set;
END_ENTITY;


ENTITY presentation_size;
 unit : presentation_size_assignment_select;
 size : planar_box;
 WHERE
  wr1 : (('AUTOMOTIVE_DESIGN.PRESENTATION_REPRESENTATION' IN TYPEOF(SELF.unit)
        ) AND item_in_context(SELF.size,SELF.unit\representation.
        context_of_items)) OR (('AUTOMOTIVE_DESIGN.AREA_IN_SET' IN TYPEOF(SELF
        .unit)) AND (SIZEOF(QUERY(ais <* SELF.unit\area_in_set.in_set.areas |
        NOT item_in_context(SELF.size,ais.area\representation.context_of_items
        ))) = 0));
END_ENTITY;


ENTITY presentation_style_assignment;
 styles : SET[1:?] OF presentation_style_select;
 WHERE
  wr1 : SIZEOF(QUERY(style1 <* SELF.styles | NOT (SIZEOF(QUERY(style2 <* (SELF
        .styles - style1) | NOT ((TYPEOF(style1) <> TYPEOF(style2)) OR (SIZEOF
        (['AUTOMOTIVE_DESIGN.' + 'SURFACE_STYLE_USAGE','AUTOMOTIVE_DESIGN.' +
        'EXTERNALLY_DEFINED_STYLE'] * TYPEOF(style1)) = 1)))) = 0))) = 0;
  wr2 : SIZEOF(QUERY(style1 <* SELF.styles |
        'AUTOMOTIVE_DESIGN.SURFACE_STYLE_USAGE' IN TYPEOF(style1))) <= 2;
END_ENTITY;


ENTITY presentation_style_by_context
 SUBTYPE OF (presentation_style_assignment);
 style_context : style_context_select;
END_ENTITY;


ENTITY presentation_view
 SUBTYPE OF (presentation_representation);
END_ENTITY;


ENTITY presentation_view_relationship
 SUBTYPE OF (representation_relationship);
 WHERE
  wr1 : ('AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN TYPEOF(SELF\
        representation_relationship.rep_1)) AND (
        'AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN TYPEOF(SELF\
        representation_relationship.rep_2));
END_ENTITY;


ENTITY presented_item
 ABSTRACT SUPERTYPE;
```

```
END_ENTITY;


ENTITY presented_item_representation;
 presentation : presentation_representation_select;
 item : presented_item;
END_ENTITY;


ENTITY prismatic_pair
 SUBTYPE OF (kinematic_pair);
END_ENTITY;


ENTITY prismatic_pair_range
 SUBTYPE OF (simple_pair_range);
 SELF\simple_pair_range.applies_to_pair : prismatic_pair;
 lower_limit_actual_translation : translational_range_measure;
 upper_limit_actual_translation : translational_range_measure;
 WHERE
  wr1 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        lower_limit_actual_translation)) OR (
        'AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        upper_limit_actual_translation))) XOR (lower_limit_actual_translation
        < upper_limit_actual_translation);
END_ENTITY;


ENTITY prismatic_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : prismatic_pair;
 actual_translation : length_measure;
END_ENTITY;


ENTITY process_change
 SUBTYPE OF (action_property);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.ACTION_RELATIONSHIP' IN TYPEOF(SELF.definition);
  wr2 : (SIZEOF(USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.ACTION_PROPERTY_REPRESENTATION.' + 'PROPERTY')) <=
        4) AND (SIZEOF(QUERY(apr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY_REPRESENTATION.PROPERTY') | NOT (apr.name IN [
        'added elements','deleted elements','relating change description',
        'related change description']))) = 0);
  wr3 : SIZEOF(QUERY(apr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY_REPRESENTATION.PROPERTY') | apr.name =
        'added elements')) <= 1;
  wr4 : SIZEOF(QUERY(apr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY_REPRESENTATION.PROPERTY') | apr.name =
        'deleted elements')) <= 1;
  wr5 : SIZEOF(QUERY(apr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY_REPRESENTATION.PROPERTY') | apr.name =
        'relating change description')) <= 1;
```

```
  wr6 : SIZEOF(QUERY(apr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY_REPRESENTATION.PROPERTY') | apr.name =
        'related change description')) <= 1;
  wr7 : SIZEOF(QUERY(apr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY_REPRESENTATION.PROPERTY') | (apr.representation.name
        = 'relating change descripiton') AND (SIZEOF(QUERY(rr <* USEDIN(apr.
        representation,'AUTOMOTIVE_DESIGN.REPRESENTATION_RELATIONSHIP.REP_1')
         | (SIZEOF(QUERY(apr_1 <* USEDIN(rr.rep_2,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY_REPRESENTATION.' + 'REPRESENTATION') | apr_1.name =
        'related change description')) = 1) AND (rr.name =
        'change description'))) = 1))) = 1;
END_ENTITY;


ENTITY process_plan
 SUBTYPE OF (product_definition_process);
 WHERE
  wr1 : SIZEOF(QUERY(ppa <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROCESS_PRODUCT_ASSOCIATION.PROCESS') | (ppa.name <> 'product output'
        ) OR NOT ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(ppa.
        defined_product)))) = 0;
  wr2 : SIZEOF(QUERY(ar <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_RELATIONSHIP.RELATED_ACTION') | NOT (
        'AUTOMOTIVE_DESIGN.PROCESS_PLAN' IN TYPEOF(ar.relating_action)))) = 0;
  wr3 : SIZEOF(QUERY(ar <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_RELATIONSHIP.RELATING_ACTION') | NOT (
        'AUTOMOTIVE_DESIGN.PROCESS_PLAN' IN TYPEOF(ar.related_action)) AND ((
        ar.name <> 'process plan operation assignment') OR (ar.related_action.
        description <> 'process operation')))) = 0;
END_ENTITY;


ENTITY process_plan_assignment
 SUBTYPE OF (action_assignment);
 items : SET[1:?] OF process_plan_item;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.PROCESS_PLAN' IN TYPEOF(SELF.assigned_action);
  wr2 : SELF.role.name = 'process plan context';
  wr3 : SIZEOF(QUERY(ar <* USEDIN(SELF.assigned_action,'AUTOMOTIVE_DESIGN.' +
        'ACTION_RELATIONSHIP.RELATING_ACTION') | (ar.name =
        'process plan operation assignment') AND (ar.related_action.
        description = 'process operation') AND (SIZEOF(QUERY(ppa <* USEDIN(ar.
        related_action,'AUTOMOTIVE_DESIGN.' +
        'PROCESS_PRODUCT_ASSOCIATION.PROCESS') | SIZEOF(QUERY(i <* SELF.items
         | i :=: ppa)) = 1)) = 1))) = 1;
  wr4 : (SIZEOF(QUERY(i <* SELF.items | NOT (
        'AUTOMOTIVE_DESIGN.PROCESS_PRODUCT_ASSOCIATION' IN TYPEOF(i)))) =
        SIZEOF(items)) OR (SIZEOF(QUERY(i <* SELF.items | (
        'AUTOMOTIVE_DESIGN.PROCESS_PRODUCT_ASSOCIATION' IN TYPEOF(i)) AND (i\
        process_product_association.process.description = 'process operation')
         AND (SIZEOF(QUERY(ar <* USEDIN(i,
        'AUTOMOTIVE_DESIGN.ACTION_RELATIONSHIP.' + 'RELATED_ACTION') | (ar.
        name = 'process plan operation assignment') AND (ar.relating_action
```

```
        :=: SELF.assigned_action))) = 1))) >= 1);
END_ENTITY;


ENTITY process_product_association;
 name : label;
 description : text;
 defined_product : characterized_product_definition;
 process : product_definition_process;
END_ENTITY;


ENTITY process_property_association;
 name : label;
 description : text;
 process : property_process;
 property_or_shape : property_or_shape_select;
END_ENTITY;


ENTITY product;
 id : identifier;
 name : label;
 description : text;
 frame_of_reference : SET[1:?] OF product_context;
END_ENTITY;


ENTITY product_category;
 name : label;
 description : OPTIONAL text;
END_ENTITY;


ENTITY product_category_relationship;
 name : label;
 description : text;
 category : product_category;
 sub_category : product_category;
 WHERE
  wr1 : acyclic_product_category_relationship(SELF,[SELF.sub_category]);
END_ENTITY;


ENTITY product_class
 SUBTYPE OF (characterized_object,product_concept);
 WHERE
  wr1 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'CONFIGURATION_ITEM.ITEM_CONCEPT')) = SIZEOF(QUERY(ci <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'CONFIGURATION_ITEM.ITEM_CONCEPT') |
        'AUTOMOTIVE_DESIGN.PRODUCT_COMPONENT' IN TYPEOF(ci))) + SIZEOF(QUERY(
        ci <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'CONFIGURATION_ITEM.ITEM_CONCEPT') | (
        'AUTOMOTIVE_DESIGN.CONFIGURABLE_ITEM' IN TYPEOF(ci)) AND (ci.
```

```
        description = 'class function definition') AND (SIZEOF(QUERY(cd <*
        USEDIN(ci,'AUTOMOTIVE_DESIGN.' + 'CONFIGURATION_DESIGN.CONFIGURATION')
         | ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION' IN TYPEOF(cd.design)) AND (
        cd.design\product_definition.frame_of_reference.name =
        'functional_definition'))) > 0))) + SIZEOF(QUERY(ci <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'CONFIGURATION_ITEM.ITEM_CONCEPT') | ci.
        description = 'product specification'));
  wr2 : SIZEOF(QUERY(ci <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'CONFIGURATION_ITEM.ITEM_CONCEPT') |
        'AUTOMOTIVE_DESIGN.PRODUCT_COMPONENT' IN TYPEOF(ci))) <= 1;
END_ENTITY;


ENTITY product_component
 SUBTYPE OF (characterized_object,configuration_item_with_context);
 WHERE
  wr1 : SELF.purpose IN ['instance required','no instance required'];
END_ENTITY;


ENTITY product_concept;
 id : identifier;
 name : label;
 description : text;
 market_context : product_concept_context;
 UNIQUE
  ur1 : id;
END_ENTITY;


ENTITY product_concept_context
 SUBTYPE OF (application_context_element);
 market_segment_type : label;
END_ENTITY;


ENTITY product_concept_feature;
 id : identifier;
 name : label;
 description : text;
END_ENTITY;


ENTITY product_concept_feature_association;
 name : label;
 description : text;
 concept : product_concept;
 feature : product_concept_feature;
END_ENTITY;


ENTITY product_concept_feature_category
 SUBTYPE OF (group);
 WHERE
```

```
  wr1 : SIZEOF(QUERY(gma <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'GROUP_ASSIGNMENT.ASSIGNED_GROUP') | (
        'AUTOMOTIVE_DESIGN.GROUP_MEMBER_ASSIGNMENT' IN TYPEOF(gma)) AND (
        SIZEOF(QUERY(gma_i <* gma.items |
        'AUTOMOTIVE_DESIGN.CONDITIONAL_CONCEPT_FEATURE' IN TYPEOF(gma_i))) > 0
        ))) = 0;
END_ENTITY;


ENTITY product_concept_feature_category_usage
 SUBTYPE OF (group_assignment);
 items : SET[1:?] OF category_usage_item;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_FEATURE_CATEGORY' IN TYPEOF(SELF.
        assigned_group);
  wr2 : SELF.role.name IN ['mandatory category usega',
        'optional category usage'];
END_ENTITY;


ENTITY product_concept_relationship;
 name : label;
 description : text;
 relating_product_concept : product_concept;
 related_product_concept : product_concept;
END_ENTITY;


ENTITY product_context
 SUBTYPE OF (application_context_element);
 discipline_type : label;
END_ENTITY;


ENTITY product_definition;
 id : identifier;
 name : label;
 description : text;
 formation : product_definition_formation;
 frame_of_reference : product_definition_context;
END_ENTITY;


ENTITY product_definition_context
 SUBTYPE OF (application_context_element);
 life_cycle_stage : label;
END_ENTITY;


ENTITY product_definition_formation;
 id : identifier;
 description : text;
 of_product : product;
 UNIQUE
```

```
  ur1 : id,of_product;
END_ENTITY;


ENTITY product_definition_formation_relationship;
 id : identifier;
 name : label;
 description : text;
 relating_product_definition_formation : product_definition_formation;
 related_product_definition_formation : product_definition_formation;
END_ENTITY;


ENTITY product_definition_occurrence
 SUBTYPE OF (product_definition);
 WHERE
  wr1 : SELF.name IN ['quantified instance','selected instance',
        'single instance'];
  wr2 : SELF.frame_of_reference.name = 'definitional template';
  wr3 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.' + 'RELATED_PRODUCT_DEFINITION') |
        'AUTOMOTIVE_DESIGN.ASSEMBLY_COMPONENT_USAGE' IN TYPEOF(pdr))) > 0) OR
        (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.' + 'RELATED_PRODUCT_DEFINITION') | (
        pdr.name = 'solution instance association') AND (
        'AUTOMOTIVE_DESIGN.SOLUTION_BASED_PRODUCT_DEFINITION' IN TYPEOF(pdr.
        relating_product_definition)))) > 0);
  wr4 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.' + 'RELATED_PRODUCT_DEFINITION') | (
        pdr.name = 'definition usage') AND (pdr.relating_product_definition.
        frame_of_reference.name = 'definitional template'))) = 1) OR (SIZEOF(
        QUERY(cd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'CONFIGURATION_DESIGN.DESIGN') |
        'AUTOMOTIVE_DESIGN.OCCURRENCE_USAGE_DEFINITION' IN TYPEOF(cd))) = 1);
  wr5 : (SELF.name <> 'quantified instance') OR ((SIZEOF(QUERY(pd <* USEDIN(
        SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION') | pd.
        description = 'occurrence quantity')) = 1) AND (SIZEOF(QUERY(pd <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION')
         | (pd.description = 'occurrence quantity') AND (SIZEOF(USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.' +
        'DEFINITION')) = 1) AND (SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | (pdr.used_representation.description = 'quantity') AND (SIZEOF(pdr
        .used_representation.items) = 1) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | ('AUTOMOTIVE_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (r_i.name =
        'quantity measure'))) = 1))) = 1))) = 1));
  wr6 : (SELF.name <> 'selected instance') OR ((SIZEOF(QUERY(pd <* USEDIN(SELF
        ,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION') | pd.
        description = 'occurrence quantity')) = 1) AND (SIZEOF(QUERY(pd <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION')
         | (pd.description = 'occurrence quantity') AND (SIZEOF(USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.' +
```

```
        'DEFINITION')) = 1) AND (SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | (pdr.used_representation.description = 'selection control') AND ({
        1<=SIZEOF(pdr.used_representation.items)<=3}) AND (SIZEOF(QUERY(r_i <*
        pdr.used_representation.items | ('AUTOMOTIVE_DESIGN.' +
        'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (r_i.name =
        'selection condition'))) = 1) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | ('AUTOMOTIVE_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (r_i.name =
        'maximum quantity'))) <= 1) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | ('AUTOMOTIVE_DESIGN.' +
        'MEASURE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (r_i.name =
        'minimum quantity'))) <= 1))) = 1))) = 1));
END_ENTITY;


ENTITY product_definition_process
 SUBTYPE OF (action);
 identification : identifier;
 INVERSE
  product_definitions : SET[1:?] OF process_product_association FOR process;
END_ENTITY;


ENTITY product_definition_relationship;
 id : identifier;
 name : label;
 description : text;
 relating_product_definition : product_definition;
 related_product_definition : product_definition;
END_ENTITY;


ENTITY product_definition_resource
 SUBTYPE OF (action_resource,product_definition);
 WHERE
  wr1 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES')) >= 1;
  wr2 : SIZEOF(QUERY(rfar <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES') | NOT (rfar.name IN [
        'reference tool','non reference toll']) OR (SIZEOF(QUERY(a <* rfar.
        operations | a.name <> 'process operation')) > 0))) = 0;
END_ENTITY;


ENTITY product_definition_shape
 SUBTYPE OF (property_definition);
 UNIQUE
  ur1 : SELF\property_definition.definition;
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.SHAPE_DEFINITION' IN TYPEOF(SELF\
        property_definition.definition));
END_ENTITY;
```

```
ENTITY product_definition_substitute;
 description : text;
 context_relationship : product_definition_relationship;
 substitute_definition : product_definition;
 WHERE
  wr1 : context_relationship.related_product_definition :<>:
        substitute_definition;
END_ENTITY;


ENTITY product_definition_usage
 SUPERTYPE OF (ONEOF(make_from_usage_option,assembly_component_usage))
 SUBTYPE OF (product_definition_relationship);
 UNIQUE
  ur1 : SELF\product_definition_relationship.id,SELF\
  product_definition_relationship.relating_product_definition,SELF\
  product_definition_relationship.related_product_definition;
 WHERE
  wr1 : acyclic_product_definition_relationship(SELF,[SELF\
        product_definition_relationship.related_product_definition],
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_USAGE.' +
        'RELATED_PRODUCT_DEFINITION');
END_ENTITY;


ENTITY product_definition_with_associated_documents
 SUBTYPE OF (product_definition);
 documentation_ids : SET[1:?] OF document;
END_ENTITY;


ENTITY product_related_product_category
 SUBTYPE OF (product_category);
 products : SET[1:?] OF product;
END_ENTITY;


ENTITY project
 SUBTYPE OF (organization);
 WHERE
  wr1 : SIZEOF(QUERY(ada <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_ASSIGNMENT.ITEMS') | ada.role.name = 'actual start')) +
        SIZEOF(QUERY(adata <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') | adata.role.name =
        'actual start')) <= 1;
  wr2 : SIZEOF(QUERY(ada <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_ASSIGNMENT.ITEMS') | ada.role.name = 'actual end')) +
        SIZEOF(QUERY(adata <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') | adata.role.name =
        'actual end')) <= 1;
  wr3 : SIZEOF(QUERY(ada <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_ASSIGNMENT.ITEMS') | ada.role.name = 'planned end')) +
        SIZEOF(QUERY(adata <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') | adata.role.name =
```

```
        'planned end')) + SIZEOF(QUERY(ea <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.'
        + 'EVENT_ASSIGNMENT.ITEMS') | ea.role.name = 'planned end')) <= 1;
  wr4 : SIZEOF(QUERY(ada <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_ASSIGNMENT.ITEMS') | ada.role.name = 'actual start')) +
        SIZEOF(QUERY(adata <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') | adata.role.name =
        'actual start')) <= 1;
  wr5 : SIZEOF(QUERY(ea <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'EVENT_ASSIGNMENT.ITEMS') | ea.role.name = 'planned start')) <= 1;
  wr6 : (SIZEOF(QUERY(ea <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'EVENT_ASSIGNMENT.ITEMS') | ea.role.name = 'actual start')) <= 1) AND
        ((SIZEOF(QUERY(ea <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'EVENT_ASSIGNMENT.ITEMS') | ea.role.name = 'actual start')) = 0) OR (
        SIZEOF(QUERY(ea <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'EVENT_ASSIGNMENT.ITEMS') | (ea.role.name = 'actual start') AND (
        'AUTOMOTIVE_DESIGN.EVENT_REFERENCE' IN TYPEOF(ea.assigned_action)) AND
         (SIZEOF(QUERY(ar <* USEDIN(ea,'AUTOMOTIVE_DESIGN.' +
        'ACTION_RELATIONSHIP.RELATING_ACTION') | (ar.name =
        'event relationship') AND ('AUTOMOTIVE_DESIGN.EVENT_REFERENCE' IN
        TYPEOF(ar.related_action)) AND (SIZEOF(QUERY(aa <* USEDIN(ar.
        related_action,'AUTOMOTIVE_DESIGN.' +
        'ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
        'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT' IN TYPEOF(aa))) > 0) AND (SIZEOF(
        QUERY(ea1 <* USEDIN(ar.related_action,
        'AUTOMOTIVE_DESIGN.ACTION_ASSIGNMENT.' + 'ASSIGNED_ACTION') | (
        'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT' IN TYPEOF(ea1)) AND (ea1.role.
        name = 'planned end') AND (SIZEOF(QUERY(ea1_i <* ea1\event_assignment.
        items | SELF :=: ea1_i)) = 1))) > 0))) > 0))) = 1));
  wr7 : SIZEOF(QUERY(aaa <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_ACTION_ASSIGNMENT.ITEMS') | aaa.role.name <> 'work program'))
         = 0;
END_ENTITY;


ENTITY project_assignment
 SUBTYPE OF (organization_assignment);
 items : SET[1:?] OF project_item;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.PROJECT' IN TYPEOF(SELF.assigned_organization);
  wr2 : SELF.role.name = 'affecting project';
END_ENTITY;


ENTITY projected_zone_definition
 SUBTYPE OF (tolerance_zone_definition);
 projection_end : shape_aspect;
 projected_length : measure_with_unit;
 WHERE
  wr1 : projected_length.value_component > 0.0;
END_ENTITY;


ENTITY projection_curve
```

```
 SUBTYPE OF (annotation_curve_occurrence);
END_ENTITY;


ENTITY projection_directed_callout
 SUBTYPE OF (draughting_callout);
 WHERE
  wr1 : SIZEOF(QUERY(p_1 <* SELF\draughting_callout.contents |
        'AUTOMOTIVE_DESIGN.PROJECTION_CURVE' IN (TYPEOF(p_1)))) = 1;
  wr2 : SIZEOF(SELF\draughting_callout.contents) >= 2;
END_ENTITY;


ENTITY promissory_usage_occurrence
 SUBTYPE OF (assembly_component_usage);
END_ENTITY;


ENTITY property_change
 SUBTYPE OF (property_definition);
 WHERE
  wr1 : SIZEOF(['AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RELATIONSHIP',
        'AUTOMOTIVE_DESIGN.SHAPE_ASPECT_RELATIONSHIP'] * TYPEOF(SELF.
        definition)) = 1;
  wr2 : (SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.' + 'RELATING_PROPERTY_DEFINITION'))
         <= 2) AND (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.' + 'RELATING_PROPERTY_DEFINITION')
         | NOT (pdr.name IN ['added property','deleted property']))) = 0);
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.' + 'RELATING_PROPERTY_DEFINITION')
         | pdr.name = 'added property')) <= 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_RELATIONSHIP.' + 'RELATING_PROPERTY_DEFINITION')
         | pdr.name = 'deleted property')) <= 1;
END_ENTITY;


ENTITY property_definition;
 name : label;
 description : text;
 definition : characterized_definition;
END_ENTITY;


ENTITY property_definition_relationship;
 name : label;
 description : text;
 relating_property_definition : property_definition;
 related_property_definition : property_definition;
END_ENTITY;


ENTITY property_definition_representation;
```

```
 name : label;
 description : text;
 definition : represented_definition;
 used_representation : representation;
END_ENTITY;


ENTITY property_process
 SUBTYPE OF (action);
 identification : identifier;
 INVERSE
  properties : SET[1:?] OF process_property_association FOR process;
END_ENTITY;


ENTITY property_value_representation
 SUBTYPE OF (representation);
 WHERE
  wr1 : SIZEOF(SELF.items) <= 3;
  wr2 : SIZEOF(QUERY(i <* SELF.items | SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM'] * TYPEOF(i)) = 0)
        ) = 0;
  wr3 : SIZEOF(QUERY(i <* SELF.items | i.name = 'lower')) <= 1;
  wr4 : SIZEOF(QUERY(i <* SELF.items | i.name = 'upper')) <= 1;
  wr5 : SIZEOF(QUERY(i <* SELF.items | i.name = 'specified value')) = 1;
END_ENTITY;


ENTITY qualified_representation_item
 SUBTYPE OF (representation_item);
 qualifiers : SET[1:?] OF value_qualifier;
 WHERE
  wr1 : SIZEOF(QUERY(temp <* qualifiers |
        'AUTOMOTIVE_DESIGN.PRECISION_QUALIFIER' IN TYPEOF(temp))) < 2;
END_ENTITY;


ENTITY qualitative_uncertainty
 SUBTYPE OF (uncertainty_qualifier);
 uncertainty_value : text;
END_ENTITY;


ENTITY quantified_assembly_component_usage
 SUBTYPE OF (assembly_component_usage);
 quantity : measure_with_unit;
END_ENTITY;


ENTITY quasi_uniform_curve
 SUBTYPE OF (b_spline_curve);
END_ENTITY;
```

```
ENTITY quasi_uniform_surface
 SUBTYPE OF (b_spline_surface);
END_ENTITY;


ENTITY rack_and_pinion_pair
 SUBTYPE OF (kinematic_pair);
 pinion_radius : length_measure;
END_ENTITY;


ENTITY rack_and_pinion_pair_range
 SUBTYPE OF (simple_pair_range);
 SELF\simple_pair_range.applies_to_pair : rack_and_pinion_pair;
 lower_limit_rack_displacement : translational_range_measure;
 upper_limit_rack_displacement : translational_range_measure;
 WHERE
  wr1 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        lower_limit_rack_displacement)) OR (
        'AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        upper_limit_rack_displacement))) XOR (lower_limit_rack_displacement <
        upper_limit_rack_displacement);
END_ENTITY;


ENTITY rack_and_pinion_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : rack_and_pinion_pair;
 actual_displacement : length_measure;
 DERIVE
  actual_rotation : plane_angle_measure :=
                    convert_plane_angle_for_pair_from_radian(SELF\pair_value.
                    applies_to_pair\kinematic_pair,(- actual_displacement /
                    SELF\pair_value.applies_to_pair\rack_and_pinion_pair.
                    pinion_radius));
END_ENTITY;


ENTITY radius_dimension
 SUBTYPE OF (dimension_curve_directed_callout);
 WHERE
  wr1 : SIZEOF(QUERY(con <* SELF.contents |
        'AUTOMOTIVE_DESIGN.PROJECTION_CURVE' IN TYPEOF(con))) <= 1;
END_ENTITY;


ENTITY ratio_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.RATIO_UNIT' IN TYPEOF(SELF\measure_with_unit.
        unit_component);
END_ENTITY;
```

```
ENTITY ratio_unit
 SUBTYPE OF (named_unit);
 WHERE
  wr1 : (SELF\named_unit.dimensions.length_exponent = 0.0) AND (SELF\
        named_unit.dimensions.mass_exponent = 0.0) AND (SELF\named_unit.
        dimensions.time_exponent = 0.0) AND (SELF\named_unit.dimensions.
        electric_current_exponent = 0.0) AND (SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 0.0) AND (SELF\named_unit.
        dimensions.amount_of_substance_exponent = 0.0) AND (SELF\named_unit.
        dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;


ENTITY rational_b_spline_curve
 SUBTYPE OF (b_spline_curve);
 weights_data : LIST[2:?] OF REAL;
 DERIVE
  weights : ARRAY[0:upper_index_on_control_points] OF REAL := list_to_array(
            weights_data,0,upper_index_on_control_points);
 WHERE
  wr1 : SIZEOF(weights_data) = SIZEOF(SELF\b_spline_curve.control_points_list)
        ;
  wr2 : curve_weights_positive(SELF);
END_ENTITY;


ENTITY rational_b_spline_surface
 SUBTYPE OF (b_spline_surface);
 weights_data : LIST[2:?] OF LIST[2:?] OF REAL;
 DERIVE
  weights : ARRAY[0:u_upper] OF ARRAY[0:v_upper] OF REAL := make_array_of_array
            (weights_data,0,u_upper,0,v_upper);
 WHERE
  wr1 : (SIZEOF(weights_data) = SIZEOF(SELF\b_spline_surface.
        control_points_list)) AND (SIZEOF(weights_data[1]) = SIZEOF(SELF\
        b_spline_surface.control_points_list[1]));
  wr2 : surface_weights_positive(SELF);
END_ENTITY;


ENTITY rectangular_closed_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1;
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND ({3<=SIZEOF(pdr.used_representation.items)
```

```
        <=4}))) <= 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | NOT (srwp_i.name IN ['orientation',
        'length','width','corner radius']))) > 0))) = 0;
  wr5 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'orientation') AND (
        'AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i)))) = 1))) = 1;
  wr6 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'width') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1;
  wr7 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'length') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1;
  wr8 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'corner radius') AND (
        SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) <= 1;
END_ENTITY;


ENTITY rectangular_composite_surface
 SUBTYPE OF (bounded_surface);
 segments : LIST[1:?] OF LIST[1:?] OF surface_patch;
 DERIVE
  n_u : INTEGER := SIZEOF(segments);
  n_v : INTEGER := SIZEOF(segments[1]);
 WHERE
  wr1 : [] = QUERY(s <* segments | n_v <> SIZEOF(s));
  wr2 : constraints_rectangular_composite_surface(SELF);
END_ENTITY;


ENTITY rectangular_pattern
```

```
SUBTYPE OF (replicate_feature);
WHERE
 wr1 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
       'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
       ) | ('AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
       used_representation)) AND (pdr.used_representation.description =
       'row layout direction'))) = 1)) = 1;
 wr2 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
       'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
       ) | ('AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
       used_representation)) AND (pdr.used_representation.description =
       'column layout direction'))) = 1)) = 1;
 wr3 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
       'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
       ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
       TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.used_representation.
       items) = 5))) = 1)) = 1;
 wr4 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
       'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
       ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
       TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
       used_representation.items | (
       'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i)) AND
        ('AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN TYPEOF(srwp_i\
       measure_representation_item.value_component)) AND (srwp_i.name =
       'number of rows'))) = 1))) = 1)) = 1;
 wr5 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
       'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
       ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
       TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
       used_representation.items | (
       'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i)) AND
        ('AUTOMOTIVE_DESIGN.COUNT_MEASURE' IN TYPEOF(srwp_i\
       measure_representation_item.value_component)) AND (srwp_i.name =
       'number of columns'))) = 1))) = 1)) = 1;
 wr6 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
       'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
       ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
       TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
       used_representation.items | (SIZEOF([
       'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
       AND (srwp_i.name = 'row spacing'))) = 1))) = 1)) = 1;
 wr7 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
       'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
```

```
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'column spacing'))) = 1))) = 1)) = 1;
END_ENTITY;


ENTITY rectangular_trimmed_surface
 SUBTYPE OF (bounded_surface);
 basis_surface : surface;
 u1 : parameter_value;
 u2 : parameter_value;
 v1 : parameter_value;
 v2 : parameter_value;
 usense : BOOLEAN;
 vsense : BOOLEAN;
 WHERE
  wr1 : u1 <> u2;
  wr2 : v1 <> v2;
  wr3 : (('AUTOMOTIVE_DESIGN.ELEMENTARY_SURFACE' IN TYPEOF(basis_surface)) AND
         (NOT ('AUTOMOTIVE_DESIGN.PLANE' IN TYPEOF(basis_surface)))) OR (
        'AUTOMOTIVE_DESIGN.SURFACE_OF_REVOLUTION' IN TYPEOF(basis_surface)) OR
         (usense = (u2 > u1));
  wr4 : (('AUTOMOTIVE_DESIGN.SPHERICAL_SURFACE' IN TYPEOF(basis_surface)) OR (
        'AUTOMOTIVE_DESIGN.TOROIDAL_SURFACE' IN TYPEOF(basis_surface))) OR (
        vsense = (v2 > v1));
END_ENTITY;


ENTITY referenced_modified_datum
 SUBTYPE OF (datum_reference);
 modifier : limit_condition;
END_ENTITY;


ENTITY relationship_condition;
 name : label;
 applicable_relationships : SET[1:?] OF relationship_with_condition;
 condition_description : text;
END_ENTITY;


ENTITY rep_item_group
 SUBTYPE OF (group,representation_item);
 WHERE
  wr1 : SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PRESENTATION_LAYER_ASSIGNMENT.ASSIGNED_ITEMS')) > 0;
  wr2 : SIZEOF(QUERY(gma <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'GROUP_ASSIGNMENT.ASSIGNED_GROUP') | (
        'AUTOMOTIVE_DESIGN.GROUP_MEMBER_ASSIGNMENT' IN TYPEOF(gma)) AND (
        SIZEOF(QUERY(gma_i <* gma.items | NOT (
        'AUTOMOTIVE_DESIGN.GEOMETRIC_REPRESENTATION_ITEM' IN TYPEOF(gma_i))))
```

```
        > 0))) = 0;
END_ENTITY;


ENTITY reparametrised_composite_curve_segment
 SUBTYPE OF (composite_curve_segment);
 param_length : parameter_value;
 WHERE
  wr1 : param_length > 0.0;
END_ENTITY;


ENTITY replicate_feature
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SIZEOF(QUERY(cor <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'CHARACTERIZED_OBJECT_RELATIONSHIP.RELATED_OBJECT') | (cor.name =
        'replicate base assignment') AND (
        'AUTOMOTIVE_DESIGN.FEATURE_DEFINITION' IN TYPEOF(cor.relating_object))
        )) = 1;
END_ENTITY;


ENTITY representation;
 id : identifier;
 name : label;
 description : text;
 items : SET[1:?] OF representation_item;
 context_of_items : representation_context;
END_ENTITY;


ENTITY representation_context;
 context_identifier : identifier;
 context_type : text;
 INVERSE
  representations_in_context : SET[1:?] OF representation FOR context_of_items
                               ;
END_ENTITY;


ENTITY representation_item;
 name : label;
 WHERE
  wr1 : SIZEOF(using_representations(SELF)) > 0;
END_ENTITY;


ENTITY representation_item_relationship;
 name : label;
 description : text;
 relating_representation_item : representation_item;
 related_representation_item : representation_item;
END_ENTITY;
```

```
ENTITY representation_map;
 mapping_origin : representation_item;
 mapped_representation : representation;
 INVERSE
  map_usage : SET[1:?] OF mapped_item FOR mapping_source;
 WHERE
  wr1 : item_in_context(SELF.mapping_origin,SELF.mapped_representation.
        context_of_items);
END_ENTITY;


ENTITY representation_relationship;
 name : label;
 description : text;
 rep_1 : representation;
 rep_2 : representation;
END_ENTITY;


ENTITY representation_relationship_with_transformation
 SUBTYPE OF (representation_relationship);
 transformation_operator : transformation;
 WHERE
  wr1 : SELF\representation_relationship.rep_1.context_of_items :<>: SELF\
        representation_relationship.rep_2.context_of_items;
END_ENTITY;


ENTITY requirement_for_action_resource
 SUBTYPE OF (action_resource_requirement);
 resources : SET[1:?] OF action_resource;
END_ENTITY;


ENTITY resource_property;
 name : label;
 description : text;
 resource : characterized_resource_definition;
END_ENTITY;


ENTITY resource_property_representation;
 name : label;
 description : text;
 property : resource_property;
 representation : representation;
END_ENTITY;


ENTITY resource_requirement_type;
 name : label;
 description : text;
END_ENTITY;
```

```
ENTITY retention
 SUBTYPE OF (action);
 WHERE
  wr1 : SIZEOF(QUERY(aa <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
        'AUTOMOTIVE_DESIGN.RETENTION_ASSIGNMENT' IN TYPEOF(aa))) > 0;
  wr2 : SIZEOF(QUERY(ada <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_ASSIGNMENT.ITEMS') | ada.role.name = 'start')) + SIZEOF(
        QUERY(adata <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') | adata.role.name = 'start')
        ) + SIZEOF(QUERY(ar <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_RELATIONSHIP.RELATING_ACTION') | ar.name = 'start')) = 1;
  wr3 : SIZEOF(QUERY(ada <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_ASSIGNMENT.ITEMS') | ada.role.name = 'earliest end')) +
        SIZEOF(QUERY(adata <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') | adata.role.name =
        'earliest end')) + SIZEOF(QUERY(ar <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.'
         + 'ACTION_RELATIONSHIP.RELATING_ACTION') | ar.name = 'earliest end'))
         + SIZEOF(QUERY(ap <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY.DEFINITION') | (ap.name = 'retention period') AND (
        SIZEOF(QUERY(apr <* USEDIN(ap,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY_REPRESENTATION.PROPERTY') | SIZEOF(QUERY(apr_r_i <*
        apr.representation.items | apr_r_i.name = 'earliest end')) > 0)) > 0))
        ) = 1;
  wr4 : SIZEOF(QUERY(ada <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_ASSIGNMENT.ITEMS') | ada.role.name = 'latest end')) +
        SIZEOF(QUERY(adata <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') | adata.role.name =
        'latest end')) + SIZEOF(QUERY(ar <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
         'ACTION_RELATIONSHIP.RELATING_ACTION') | ar.name = 'latest end')) +
        SIZEOF(QUERY(ap <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY.DEFINITION') | (ap.name = 'retention period') AND (
        SIZEOF(QUERY(apr <* USEDIN(ap,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY_REPRESENTATION.PROPERTY') | SIZEOF(QUERY(apr_r_i <*
        apr.representation.items | apr_r_i.name = 'latest end')) > 0)) > 0)))
        = 1;
  wr5 : (SIZEOF(QUERY(ap <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY.DEFINITION') | ap.name = 'retention period')) <= 1)
        AND ((SIZEOF(QUERY(ap <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY.DEFINITION') | ap.name = 'retention period')) = 0) OR
         (SIZEOF(QUERY(ap <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY.DEFINITION') | (ap.name = 'retention period') AND (
        SIZEOF(USEDIN(ap,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY_REPRESENTATION.PROPERTY')) = 1) AND (SIZEOF(QUERY(apr
        <* USEDIN(ap,'AUTOMOTIVE_DESIGN.' +
        'ACTION_PROPERTY_REPRESENTATION.PROPERTY') | (SIZEOF(apr.
        representation.items) <= 2) AND (SIZEOF(QUERY(apr_r_i <* apr.
        representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.TIME_MEASURE_WITH_UNIT'] * TYPEOF(apr_r_i)) = 2)
        AND (apr_r_i.name IN ['earliest end','latest end']))) = SIZEOF(apr.
        representation.items)) AND (SIZEOF(QUERY(apr_r_i <* apr.representation
```

```
        .items | apr_r_i.name = 'earliest end')) <= 1) AND (SIZEOF(QUERY(
        apr_r_i <* apr.representation.items | apr_r_i.name = 'latest end')) <=
         1))) = 1))) = 1));
  wr6 : (SIZEOF(USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_RELATIONSHIP.RELATED_ACTION')) = 0) AND (SIZEOF(USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'ACTION_RELATIONSHIP.RELATING_ACTION')) <= 3)
        AND (SIZEOF(QUERY(ar <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_RELATIONSHIP.RELATING_ACTION') | (ar.name IN ['start',
        'earliest end','latest end']) AND ('AUTOMOTIVE_DESIGN.EVENT_REFERENCE'
         IN TYPEOF(ar.related_action)))) = SIZEOF(USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'ACTION_RELATIONSHIP.RELATING_ACTION'))) AND (
        SIZEOF(QUERY(ar <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_RELATIONSHIP.RELATING_ACTION') | ar.name = 'start')) <= 1) AND
         (SIZEOF(QUERY(ar <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_RELATIONSHIP.RELATING_ACTION') | ar.name = 'earliest end')) <=
         1) AND (SIZEOF(QUERY(ar <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'ACTION_RELATIONSHIP.RELATING_ACTION') | ar.name = 'latest end')) <= 1
        );
END_ENTITY;


ENTITY retention_assignment
 SUBTYPE OF (action_assignment);
 items : SET[1:?] OF retention_item;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.RETENTION' IN TYPEOF(SELF.assigned_action);
  wr2 : SELF.role.name = 'retention';
END_ENTITY;


ENTITY revolute_pair
 SUBTYPE OF (kinematic_pair);
END_ENTITY;


ENTITY revolute_pair_range
 SUBTYPE OF (simple_pair_range);
 SELF\simple_pair_range.applies_to_pair : revolute_pair;
 lower_limit_actual_rotation : rotational_range_measure;
 upper_limit_actual_rotation : rotational_range_measure;
 WHERE
  wr1 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        lower_limit_actual_rotation)) OR ('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE'
        IN TYPEOF(upper_limit_actual_rotation))) XOR (
        lower_limit_actual_rotation < upper_limit_actual_rotation);
END_ENTITY;


ENTITY revolute_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : revolute_pair;
 actual_rotation : plane_angle_measure;
END_ENTITY;
```

```
ENTITY revolved_face_solid
 SUBTYPE OF (swept_face_solid);
 axis : axis1_placement;
 angle : plane_angle_measure;
 DERIVE
  axis_line : line := line(axis.location,vector(axis.z,1.0));
END_ENTITY;


ENTITY rib
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND ({1<=SIZEOF(pdr.
        used_representation.items)<=3}))) = 1)) = 1;
  wr2 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | NOT (srwp_i.name IN ['orientation',
        'fillet radius']))) > 0))) > 0)) = 0;
  wr3 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'fillet radius'))) = 1))) <= 1)) <= 1;
  wr4 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'shoulder radius'))) = 1))) <= 1)) <= 1;
  wr5 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'rib profile') AND (sa.
        description = 'open profile occurrence') AND (SIZEOF(QUERY(sar <*
        USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar))) = 1)
        AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
```

```
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar)) AND (
        sar.name = 'open profile usage') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE',
        'AUTOMOTIVE_DESIGN.PARTIAL_CIRCULAR_PROFILE',
        'AUTOMOTIVE_DESIGN.ROUNDED_U_PROFILE','AUTOMOTIVE_DESIGN.VEE_PROFILE',
        'AUTOMOTIVE_DESIGN.TEE_PROFILE','AUTOMOTIVE_DESIGN.OPEN_PATH_PROFILE']
         * TYPEOF(sar.relating_shape_aspect)) = 1))) = 1))) = 1))) = 1;
  wr6 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'sweep path') AND (sa.
        description = 'path feature component occurrence') AND (SIZEOF(QUERY(
        sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar))) = 1)
        AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar)) AND (
        sar.name = 'path feature component usage') AND (
        'AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN TYPEOF(sar.
        relating_shape_aspect)))) = 1))) = 1))) = 1;
  wr7 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'taper') AND (sa.description =
        'taper occurrence') AND (SIZEOF(QUERY(sar <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar))) = 1)
        AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar)) AND (
        sar.name = 'taper usage') AND ('AUTOMOTIVE_DESIGN.TAPER' IN TYPEOF(sar
        .relating_shape_aspect)) AND (sar.relating_shape_aspect.description =
        'directed taper'))) = 1))) = 1))) = 1;
END_ENTITY;


ENTITY right_angular_wedge
 SUBTYPE OF (geometric_representation_item);
 position : axis2_placement_3d;
 x : positive_length_measure;
 y : positive_length_measure;
 z : positive_length_measure;
 ltx : length_measure;
 WHERE
  wr1 : ((0.0 <= ltx) AND (ltx < x));
END_ENTITY;
```

```
ENTITY right_circular_cone
 SUBTYPE OF (geometric_representation_item);
 position : axis1_placement;
 height : positive_length_measure;
 radius : length_measure;
 semi_angle : plane_angle_measure;
 WHERE
  wr1 : radius >= 0.0;
END_ENTITY;


ENTITY right_circular_cylinder
 SUBTYPE OF (geometric_representation_item);
 position : axis1_placement;
 height : positive_length_measure;
 radius : positive_length_measure;
END_ENTITY;


ENTITY rolling_curve_pair
 SUBTYPE OF (planar_curve_pair);
END_ENTITY;


ENTITY rolling_curve_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : rolling_curve_pair;
 actual_point_on_curve_1 : point_on_curve;
 WHERE
  wr1 : SELF\pair_value.applies_to_pair\planar_curve_pair.curve_1 :=:
        actual_point_on_curve_1.basis_curve;
END_ENTITY;


ENTITY rolling_surface_pair
 SUBTYPE OF (surface_pair);
END_ENTITY;


ENTITY rolling_surface_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : rolling_surface_pair;
 actual_point_on_surface : point_on_surface;
 actual_rotation : plane_angle_measure;
 WHERE
  wr1 : SELF\pair_value.applies_to_pair\surface_pair.surface_1 :=:
        actual_point_on_surface.basis_surface;
END_ENTITY;


ENTITY rotation_about_direction;
 direction_of_axis : direction;
 rotation_angle : plane_angle_measure;
 WHERE
```

```
  wr1 : SIZEOF(direction_of_axis.direction_ratios) = 3;
END_ENTITY;


ENTITY round_hole
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'boundary') AND (sa.description
        = 'closed profile occurrence') AND (SIZEOF(QUERY(sar <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar))) = 1)
        AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar)) AND (
        sar.name = 'closed profile usage') AND (
        'AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE' IN TYPEOF(sar.
        relating_shape_aspect)))) = 1))) = 1))) = 1;
  wr2 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'sweep path') AND (sa.
        description = 'path feature component occurrence') AND (SIZEOF(QUERY(
        sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar))) = 1)
        AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar)) AND (
        sar.name = 'path feature component usage') AND (
        'AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN TYPEOF(sar.
        relating_shape_aspect)) AND (sar.relating_shape_aspect.description =
        'linear'))) = 1))) = 1))) = 1;
  wr3 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'bottom condition') AND (sa.
        description = 'hole bottom occurrence') AND (SIZEOF(QUERY(sar <* USEDIN
        (sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar))) =
        1) AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)) AND
         (sar.name = 'hole bottom usage') AND ('AUTOMOTIVE_DESIGN.HOLE_BOTTOM'
         IN TYPEOF(sar.relating_shape_aspect)) AND (sar.relating_shape_aspect.
        description IN ['conical','flat','flat with radius','spherical',
```

```
        'through','flat with chamfer'])))) = 1))) = 1))) = 1;
  wr4 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'taper') AND (sa.description =
        'taper occurrence') AND (SIZEOF(QUERY(sar <* USEDIN(sa,
        'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar))) =
        1) AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)) AND
         (sar.name = 'taper usage') AND ('AUTOMOTIVE_DESIGN.TAPER' IN TYPEOF(
        sar.relating_shape_aspect)))) = 1))) <= 1))) <= 1;
  wr5 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
        SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'top chamfer') AND (SIZEOF(QUERY
        (sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar))) =
        2) AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)) AND
         (sar.name = 'taper usage') AND ('AUTOMOTIVE_DESIGN.CHAMFER_OFFSET' IN
         TYPEOF(sar.related_shape_aspect)) AND (sar.related_shape_aspect.name
        IN ['first chamfer offset','second chamfer offset']))) = 2))) <= 1)))
        <= 1;
END_ENTITY;


ENTITY rounded_u_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1;
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(pdr.used_representation.items) = 2)
        )) <= 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'orientation') AND (
        'AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i)))) = 1))) = 1;
```

```
  wr5 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'width') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1;
END_ENTITY;


ENTITY runout_zone_definition
 SUBTYPE OF (tolerance_zone_definition);
 orientation : runout_zone_orientation;
END_ENTITY;


ENTITY runout_zone_orientation;
 angle : measure_with_unit;
END_ENTITY;


ENTITY runout_zone_orientation_reference_direction
 SUBTYPE OF (runout_zone_orientation);
 orientation_defining_relationship : shape_aspect_relationship;
END_ENTITY;


ENTITY screw_pair
 SUBTYPE OF (kinematic_pair);
 pitch : length_measure;
END_ENTITY;


ENTITY screw_pair_range
 SUBTYPE OF (simple_pair_range);
 SELF\simple_pair_range.applies_to_pair : screw_pair;
 lower_limit_actual_rotation : rotational_range_measure;
 upper_limit_actual_rotation : rotational_range_measure;
 WHERE
  wr1 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        lower_limit_actual_rotation)) OR ('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE'
        IN TYPEOF(upper_limit_actual_rotation))) XOR (
        lower_limit_actual_rotation < upper_limit_actual_rotation);
END_ENTITY;


ENTITY screw_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : screw_pair;
 actual_rotation : plane_angle_measure;
 DERIVE
  actual_translation : length_measure := SELF\pair_value.applies_to_pair\
                       screw_pair.pitch * plane_angle_for_pair_in_radian(SELF\
```

```
                        pair_value.applies_to_pair,actual_rotation) / (2 * PI);
END_ENTITY;

ENTITY seam_curve
 SUBTYPE OF (surface_curve);
 WHERE
  wr1 : SIZEOF(SELF\surface_curve.associated_geometry) = 2;
  wr2 : associated_surface(SELF\surface_curve.associated_geometry[1]) =
        associated_surface(SELF\surface_curve.associated_geometry[2]);
  wr3 : 'AUTOMOTIVE_DESIGN.PCURVE' IN TYPEOF(SELF\surface_curve.
        associated_geometry[1]);
  wr4 : 'AUTOMOTIVE_DESIGN.PCURVE' IN TYPEOF(SELF\surface_curve.
        associated_geometry[2]);
END_ENTITY;

ENTITY security_classification;
 name : label;
 purpose : text;
 security_level : security_classification_level;
END_ENTITY;

ENTITY security_classification_assignment
 ABSTRACT SUPERTYPE;
 assigned_security_classification : security_classification;
 role : security_classification_role;
END_ENTITY;

ENTITY security_classification_level;
 name : label;
END_ENTITY;

ENTITY security_classification_role;
 name : label;
 description : text;
END_ENTITY;

ENTITY serial_numbered_effectivity
 SUBTYPE OF (effectivity);
 effectivity_start_id : identifier;
 effectivity_end_id : OPTIONAL identifier;
END_ENTITY;

ENTITY shape_aspect;
 name : label;
 description : text;
 of_shape : product_definition_shape;
 product_definitional : LOGICAL;
END_ENTITY;
```

```
ENTITY shape_aspect_associativity
 SUBTYPE OF (shape_aspect_relationship);
 WHERE
  wr1 : SELF.relating_shape_aspect.product_definitional;
  wr2 : NOT (SELF.related_shape_aspect.product_definitional);
  wr3 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | NOT (SIZEOF(QUERY(sdr <* QUERY(pdr
        <* USEDIN(pd,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINITION_REPRESENTATION' IN TYPEOF(pdr)))
         | NOT (SIZEOF(QUERY(it <* sdr.used_representation.items | NOT (
        'AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF(it)) AND (SIZEOF(QUERY(aco <*
        USEDIN(it,'AUTOMOTIVE_DESIGN.STYLED_ITEM.ITEM') | SIZEOF(TYPEOF(aco) *
         ['AUTOMOTIVE_DESIGN.LEADER_CURVE',
        'AUTOMOTIVE_DESIGN.PROJECTION_CURVE']) = 1)) + SIZEOF(USEDIN(it,
        'AUTOMOTIVE_DESIGN.' + 'ANNOTATION_FILL_AREA.BOUNDARIES')) >= 1))) = 1
        ))) = 0))) = 0;
  wr4 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | NOT (SIZEOF(QUERY(sdr <* QUERY(pdr
        <* USEDIN(pd,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINITION_REPRESENTATION' IN TYPEOF(pdr)))
         | NOT (check_associative_shape_aspects(sdr)))) = 0))) = 0;
END_ENTITY;


ENTITY shape_aspect_deriving_relationship
 SUBTYPE OF (shape_aspect_relationship);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.DERIVED_SHAPE_ASPECT' IN TYPEOF(SELF\
        shape_aspect_relationship.relating_shape_aspect);
END_ENTITY;


ENTITY shape_aspect_relationship;
 name : label;
 description : text;
 relating_shape_aspect : shape_aspect;
 related_shape_aspect : shape_aspect;
END_ENTITY;


ENTITY shape_aspect_transition
 SUBTYPE OF (shape_aspect_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(pdr_1 <* USEDIN(SELF.relating_shape_aspect,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | SIZEOF(QUERY(pdr_2 <* USEDIN(SELF.related_shape_aspect,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | SIZEOF(QUERY(item_1 <* pdr_1.used_representation.items | SIZEOF(
        QUERY(item_2 <* pdr_2.used_representation.items | NOT (NOT ((
        'AUTOMOTIVE_DESIGN.EDGE' IN TYPEOF(item_1))) XOR (
        'AUTOMOTIVE_DESIGN.EDGE' IN TYPEOF(item_2))) OR NOT (NOT ((
```

```
        'AUTOMOTIVE_DESIGN.FACE' IN TYPEOF(item_1))) XOR (
        'AUTOMOTIVE_DESIGN.FACE' IN TYPEOF(item_2))) OR ((
        'AUTOMOTIVE_DESIGN.FACE' IN TYPEOF(item_1)) AND (
        'AUTOMOTIVE_DESIGN.EDGE' IN TYPEOF(item_2))))) > 0)) > 0)) > 0)) = 0;
  wr2 : SIZEOF(QUERY(pdr_1 <* USEDIN(SELF.relating_shape_aspect,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | SIZEOF(QUERY(pdr_2 <* USEDIN(SELF.related_shape_aspect,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | SIZEOF(QUERY(item_1 <* pdr_1.used_representation.items | (
        'AUTOMOTIVE_DESIGN.EDGE' IN TYPEOF(item_1)) AND (SIZEOF(QUERY(item_2 <*
        pdr_2.used_representation.items | ('AUTOMOTIVE_DESIGN.EDGE' IN TYPEOF(
        item_2)) AND NOT ((item_1\edge.edge_start :=: item_2\edge.edge_start)
        OR (item_1\edge.edge_start :=: item_2\edge.edge_end) OR (item_1\edge.
        edge_end :=: item_2\edge.edge_start) OR (item_1\edge.edge_end :=:
        item_2\edge.edge_end)))) > 0))) > 0)) > 0)) = 0;
  wr3 : SIZEOF(QUERY(pdr_1 <* USEDIN(SELF.relating_shape_aspect,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | SIZEOF(QUERY(pdr_2 <* USEDIN(SELF.related_shape_aspect,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | SIZEOF(QUERY(item_1 <* pdr_1.used_representation.items | (
        'AUTOMOTIVE_DESIGN.FACE' IN TYPEOF(item_1)) AND (SIZEOF(QUERY(item_2 <*
        pdr_2.used_representation.items | ('AUTOMOTIVE_DESIGN.FACE' IN TYPEOF(
        item_2)) AND (SIZEOF(QUERY(f_1 <* item_1\face.bounds | (
        'AUTOMOTIVE_DESIGN.EDGE_LOOP' IN TYPEOF(f_1.bound)) AND (SIZEOF(QUERY(
        f_2 <* item_2\face.bounds | ('AUTOMOTIVE_DESIGN.EDGE_LOOP' IN TYPEOF(
        f_2.bound)) AND (SIZEOF(QUERY(oe_1 <* f_1.bound\path.edge_list |
        SIZEOF(QUERY(oe_2 <* f_2.bound\path.edge_list | oe_1.edge_element :=:
        oe_2.edge_element)) > 0)) > 0))) > 0))) > 0))) = 0))) > 0)) > 0)) = 0;
  wr4 : SELF.name IN ['g0','g1','g2'];
END_ENTITY;


ENTITY shape_defining_relationship
 SUBTYPE OF (shape_aspect_relationship);
END_ENTITY;


ENTITY shape_definition_representation
 SUBTYPE OF (property_definition_representation);
 WHERE
  wr1 : ('AUTOMOTIVE_DESIGN.SHAPE_DEFINITION' IN TYPEOF(SELF.definition.
        definition)) OR ('AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(SELF.definition));
  wr2 : 'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN TYPEOF(SELF.
        used_representation);
END_ENTITY;


ENTITY shape_dimension_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(temp <* SELF.items | NOT (
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(temp)))) = 0
```

```
        ;
  wr2 : SIZEOF(SELF.items) <= 2;
  wr3 : SIZEOF(QUERY(pos_mri <* QUERY(real_mri <* SELF.items | 'REAL' IN
        TYPEOF(real_mri\measure_with_unit.value_component)) | NOT (pos_mri\
        measure_with_unit.value_component > 0.0))) = 0;
END_ENTITY;


ENTITY shape_representation
 SUBTYPE OF (representation);
END_ENTITY;


ENTITY shape_representation_relationship
 SUBTYPE OF (representation_relationship);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN (TYPEOF(SELF\
        representation_relationship.rep_1) + TYPEOF(SELF\
        representation_relationship.rep_2));
END_ENTITY;


ENTITY shape_representation_with_parameters
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(i <* SELF.items | SIZEOF(['AUTOMOTIVE_DESIGN.PLACEMENT',
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM'] * TYPEOF(i)) = 1)
        ) = SIZEOF(SELF.items);
END_ENTITY;


ENTITY shell_based_surface_model
 SUBTYPE OF (geometric_representation_item);
 sbsm_boundary : SET[1:?] OF shell;
 WHERE
  wr1 : constraints_geometry_shell_based_surface_model(SELF);
END_ENTITY;


ENTITY si_unit
 SUBTYPE OF (named_unit);
 prefix : OPTIONAL si_prefix;
 name : si_unit_name;
 DERIVE
  SELF\named_unit.dimensions : dimensional_exponents := dimensions_for_si_unit
                               (SELF.name);
END_ENTITY;


ENTITY simple_pair_range;
 applies_to_pair : kinematic_pair;
END_ENTITY;
```

```
ENTITY sliding_curve_pair
 SUBTYPE OF (planar_curve_pair);
END_ENTITY;


ENTITY sliding_curve_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : sliding_curve_pair;
 actual_point_on_curve_1 : point_on_curve;
 actual_point_on_curve_2 : point_on_curve;
 WHERE
  wr1 : SELF\pair_value.applies_to_pair\planar_curve_pair.curve_1 :=:
        actual_point_on_curve_1.basis_curve;
  wr2 : SELF\pair_value.applies_to_pair\planar_curve_pair.curve_2 :=:
        actual_point_on_curve_2.basis_curve;
END_ENTITY;


ENTITY sliding_surface_pair
 SUBTYPE OF (surface_pair);
END_ENTITY;


ENTITY sliding_surface_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : sliding_surface_pair;
 actual_point_on_surface_1 : point_on_surface;
 actual_point_on_surface_2 : point_on_surface;
 actual_rotation : plane_angle_measure;
 WHERE
  wr1 : SELF\pair_value.applies_to_pair\surface_pair.surface_1 :=:
        actual_point_on_surface_1.basis_surface;
  wr2 : SELF\pair_value.applies_to_pair\surface_pair.surface_2 :=:
        actual_point_on_surface_2.basis_surface;
END_ENTITY;


ENTITY slot
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND ({1<=SIZEOF(pdr.
        used_representation.items)<=2}))) = 1)) = 1;
  wr2 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (pdr.used_representation.name =
        'depth') AND (SIZEOF(QUERY(srwp_i <* pdr.used_representation.items | (
        SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
```

```
       = 1))) <= 1)) <= 1;
wr3 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
      SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'sweep shape') AND (sa.
      description = 'open profile occurrence') AND (SIZEOF(QUERY(sar <*
      USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar))) = 1)
      AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
      'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar)) AND (
      sar.name = 'open profile usage') AND (SIZEOF([
      'AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE',
      'AUTOMOTIVE_DESIGN.PARTIAL_CIRCULAR_PROFILE',
      'AUTOMOTIVE_DESIGN.ROUNDED_U_PROFILE','AUTOMOTIVE_DESIGN.VEE_PROFILE',
      'AUTOMOTIVE_DESIGN.TEE_PROFILE','AUTOMOTIVE_DESIGN.OPEN_PATH_PROFILE']
       * TYPEOF(sar.relating_shape_aspect)) = 1))) = 1))) = 1))) = 1;
wr4 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
      SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'sweep path') AND (sa.
      description = 'path feature component occurrence') AND (SIZEOF(QUERY(
      sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar))) = 1)
      AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
      'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar)) AND (
      sar.name = 'path feature component usage') AND (
      'AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT' IN TYPEOF(sar.
      relating_shape_aspect)))) = 1))) = 1))) = 1;
wr5 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
      SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'end condition at path start')
      AND (sa.description = 'slot end occurrence') AND (SIZEOF(QUERY(sar <*
      USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar))) =
      1) AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
      'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)) AND
       (sar.name = 'slot end usage') AND ('AUTOMOTIVE_DESIGN.SLOT_END' IN
      TYPEOF(sar.relating_shape_aspect)))) = 1))) = 1))) = 1;
wr6 : SIZEOF(QUERY(pds <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds)) AND (
      SIZEOF(QUERY(sa <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
```

```
        'SHAPE_ASPECT.OF_SHAPE') | (sa.name = 'end condition at path end') AND
         (sa.description = 'slot end occurrence') AND (SIZEOF(QUERY(sar <*
        USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar))) =
        1) AND (SIZEOF(QUERY(sar <* USEDIN(sa,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | (
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)) AND
         (sar.name = 'slot end usage') AND ('AUTOMOTIVE_DESIGN.SLOT_END' IN
        TYPEOF(sar.relating_shape_aspect)))) = 1))) = 1))) = 1;
END_ENTITY;


ENTITY slot_end
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SELF.description IN ['open','radiused','flat','woodruff'];
  wr3 : NOT (SELF.description IN ['open','radiused']) OR (SIZEOF(QUERY(pdr <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 0);
  wr4 : NOT (SELF.description IN ['flat','woodruff']) OR (SIZEOF(QUERY(pdr <*
        USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1);
  wr5 : (SELF.description <> 'flat') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.used_representation.
        items) = 2))) = 1);
  wr6 : (SELF.description <> 'flat') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'first radius') AND (SIZEOF
        (['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1);
  wr7 : (SELF.description <> 'flat') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'second radius') AND (
        SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1);
  wr8 : (SELF.description <> 'woodruff') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
```

```
        TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.used_representation.
        items) = 1))) = 1);
  wr9 : (SELF.description <> ' woodruff ') OR (SIZEOF(QUERY(pdr <* USEDIN(SELF
        ,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'radius') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1);
  wr10 : SIZEOF(QUERY(fcr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
         'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (
         'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(fcr))
         AND (fcr.description = 'slot end usage') AND (
         'AUTOMOTIVE_DESIGN.SLOT' IN TYPEOF(fcr.related_shape_aspect)) AND (
         fcr.related_shape_aspect.name IN ['end condition at path start',
         'end condition at path end']) AND (fcr.related_shape_aspect.
         description = 'slot end occurrence'))) = 1;
END_ENTITY;


ENTITY solid_angle_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.SOLID_ANGLE_UNIT' IN TYPEOF(SELF\measure_with_unit.
        unit_component);
END_ENTITY;


ENTITY solid_angle_unit
 SUBTYPE OF (named_unit);
 WHERE
  wr1 : (SELF\named_unit.dimensions.length_exponent = 0.0) AND (SELF\
        named_unit.dimensions.mass_exponent = 0.0) AND (SELF\named_unit.
        dimensions.time_exponent = 0.0) AND (SELF\named_unit.dimensions.
        electric_current_exponent = 0.0) AND (SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 0.0) AND (SELF\named_unit.
        dimensions.amount_of_substance_exponent = 0.0) AND (SELF\named_unit.
        dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;


ENTITY solid_model
 SUPERTYPE OF (ONEOF(csg_solid,manifold_solid_brep,swept_face_solid,
               solid_replica))
 SUBTYPE OF (geometric_representation_item);
END_ENTITY;


ENTITY solid_replica
 SUBTYPE OF (solid_model);
 parent_solid : solid_model;
 transformation : cartesian_transformation_operator_3d;
```

```
 WHERE
  wr1 : acyclic_solid_replica(SELF,parent_solid);
END_ENTITY;


ENTITY solution_alternative_definition
 SUBTYPE OF (configuration_design);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.SOLUTION_BASED_PRODUCT_DEFINITION' IN TYPEOF(SELF.
        design);
  wr2 : 'AUTOMOTIVE_DESIGN.PRODUCT_COMPONENT' IN TYPEOF(SELF.configuration);
END_ENTITY;


ENTITY solution_based_product_definition
 SUBTYPE OF (product_definition);
 WHERE
  wr1 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') | (pdr.
        name = 'solution alternative definition') AND (
        'AUTOMOTIVE_DESIGN.SOLUTION_BASED_PRODUCT_DEFINITION' IN TYPEOF(pdr.
        relating_product_definition)))) + SIZEOF(QUERY(cd <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'CONFIGURATION_DESIGN.DESIGN') |
        'AUTOMOTIVE_DESIGN.SOLUTION_ALTERNATIVE_DEFINITION' IN TYPEOF(cd)))) =
         1;
  wr2 : SELF.name IN ['final','supplier','technical','technical supplier',
        'technical final','supplier final','technical supplier final'];
  wr3 : NOT (SELF.name IN ['supplier','technical supplier','supplier final',
        'technical supplier final']) OR (SIZEOF(QUERY(aoa <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | aoa.
        role.name = 'supplier')) = 1);
  wr4 : NOT (SELF.name IN ['final','technical final','supplier final',
        'technical supplier final']) OR (SIZEOF(QUERY(pd <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(
        QUERY(pdr <* USEDIN(pd,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.' + 'DEFINITION') | (pdr.
        used_representation.description = 'final item characteristics') AND (
        SIZEOF(QUERY(r_i <* pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.' + 'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i
        )) AND (r_i.name = 'final item status'))) = 1))) = 1)) = 1);
  wr5 : NOT (SELF.name IN ['final','technical final','supplier final',
        'technical supplier final']) OR (SIZEOF(QUERY(pd <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(
        QUERY(pdr <* USEDIN(pd,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_RELATIONSHIP.' + 'RELATING_PRODUCT_DEFINITION') |
        (pdr.name = 'final specification') AND (pdr.related_product_definition
        .frame_of_reference.name = 'definitional template'))) = 1)) >= 1) OR (
        SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.' +
        'DEFINITION') | (pdr.used_representation.description =
        'final item characteristics') AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | ('AUTOMOTIVE_DESIGN.' +
```

```
        'DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (r_i.name =
        'final specification'))) >= 1))) = 1)) > 0);
END_ENTITY;


ENTITY specified_higher_usage_occurrence
 SUBTYPE OF (assembly_component_usage);
 upper_usage : assembly_component_usage;
 next_usage : next_assembly_usage_occurrence;
 UNIQUE
  ur1 : upper_usage,next_usage;
 WHERE
  wr1 : SELF :<>: upper_usage;
  wr2 : SELF\product_definition_relationship.relating_product_definition :=:
        upper_usage.relating_product_definition;
  wr3 : SELF\product_definition_relationship.related_product_definition :=:
        next_usage.related_product_definition;
  wr4 : upper_usage.related_product_definition :=: next_usage.
        relating_product_definition;
  wr5 : NOT ('AUTOMOTIVE_DESIGN.' + 'PROMISSORY_USAGE_OCCURRENCE' IN TYPEOF(
        upper_usage));
END_ENTITY;


ENTITY sphere
 SUBTYPE OF (geometric_representation_item);
 radius : positive_length_measure;
 centre : point;
END_ENTITY;


ENTITY spherical_pair
 SUBTYPE OF (kinematic_pair);
END_ENTITY;


ENTITY spherical_pair_range
 SUBTYPE OF (simple_pair_range);
 SELF\simple_pair_range.applies_to_pair : spherical_pair;
 lower_limit_yaw : rotational_range_measure;
 upper_limit_yaw : rotational_range_measure;
 lower_limit_pitch : rotational_range_measure;
 upper_limit_pitch : rotational_range_measure;
 lower_limit_roll : rotational_range_measure;
 upper_limit_roll : rotational_range_measure;
 WHERE
  wr1 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(lower_limit_yaw)) OR (
        'AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(upper_limit_yaw))) XOR (
        lower_limit_yaw < upper_limit_yaw);
  wr2 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(lower_limit_pitch)) OR
         ('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(upper_limit_pitch)))
        XOR (lower_limit_pitch < upper_limit_pitch);
  wr3 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(lower_limit_roll)) OR
        ('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(upper_limit_roll))) XOR
```

```
         (lower_limit_roll < upper_limit_roll);
END_ENTITY;


ENTITY spherical_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : spherical_pair;
 input_orientation : spatial_rotation;
 DERIVE
  actual_orientation : ypr_rotation := convert_spatial_to_ypr_rotation(SELF\
                       pair_value.applies_to_pair,input_orientation);
END_ENTITY;


ENTITY spherical_surface
 SUBTYPE OF (elementary_surface);
 radius : positive_length_measure;
END_ENTITY;


ENTITY square_u_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1;
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND ({4<=SIZEOF(pdr.used_representation.items)
        <=6}))) = 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | NOT (srwp_i.name IN ['orientation','width'
        ,'first angle','second angle','first radius','second radius']))) > 0))
        ) = 0;
  wr5 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'orientation') AND (
        'AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i)))) = 1))) = 1;
  wr6 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'width') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
```

```
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) = 1;
  wr7 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'first angle') AND (SIZEOF(
        ['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) =
         2))) = 1))) = 1;
  wr8 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'second angle') AND (SIZEOF
        (['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) =
         2))) = 1))) = 1;
  wr9 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'first radius') AND (SIZEOF
        (['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
         = 1))) <= 1;
  wr10 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
         'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
         pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
         used_representation.items | (srwp_i.name = 'second radius') AND (
         SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2))
         ) = 1))) <= 1;
END_ENTITY;


ENTITY standard_uncertainty
 SUBTYPE OF (uncertainty_qualifier);
 uncertainty_value : REAL;
END_ENTITY;


ENTITY straightness_tolerance
 SUBTYPE OF (geometric_tolerance);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE'
         IN TYPEOF(SELF));
END_ENTITY;


ENTITY structured_dimension_callout
 SUBTYPE OF (draughting_callout);
```

```
 WHERE
  wr1 : SIZEOF(TYPEOF(SELF) * ['AUTOMOTIVE_DESIGN.DATUM_FEATURE_CALLOUT',
        'AUTOMOTIVE_DESIGN.DATUM_TARGET_CALLOUT',
        'AUTOMOTIVE_DESIGN.GEOMETRICAL_TOLERANCE_CALLOUT',
        'AUTOMOTIVE_DESIGN.LEADER_DIRECTED_CALLOUT',
        'AUTOMOTIVE_DESIGN.PROJECTION_DIRECTED_CALLOUT',
        'AUTOMOTIVE_DESIGN.DIMENSION_CURVE_DIRECTED_CALLOUT']) = 0;
  wr2 : SIZEOF(QUERY(ato <* QUERY(con <* SELF.contents | (
        'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(con))) | NOT (
        ato.name IN ['dimension value','tolerance value','unit text',
        'prefix text','suffix text']))) = 0;
  wr3 : SIZEOF(QUERY(ato <* QUERY(con <* SELF.contents | (
        'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(con))) | (ato
        .name = 'dimension value'))) >= 1;
  wr4 : SIZEOF(QUERY(dcr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'DIMENSION_CALLOUT_COMPONENT_RELATIONSHIP.' +
        'RELATING_DRAUGHTING_CALLOUT') | dcr.name = 'prefix')) <= 1;
  wr5 : SIZEOF(QUERY(dcr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'DIMENSION_CALLOUT_COMPONENT_RELATIONSHIP.' +
        'RELATING_DRAUGHTING_CALLOUT') | dcr.name = 'suffix')) <= 1;
  wr6 : ((SIZEOF(QUERY(ato <* QUERY(con <* SELF.contents | (
        'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(con))) | (ato
        .name = 'prefix text'))) > 0) AND (SIZEOF(QUERY(dcr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'DIMENSION_CALLOUT_COMPONENT_RELATIONSHIP.' +
        'RELATING_DRAUGHTING_CALLOUT') | dcr.name = 'prefix')) = 1));
  wr7 : ((SIZEOF(QUERY(ato <* QUERY(con <* SELF.contents | (
        'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_OCCURRENCE' IN TYPEOF(con))) | (ato
        .name = 'suffix text'))) > 0) AND (SIZEOF(QUERY(dcr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'DIMENSION_CALLOUT_COMPONENT_RELATIONSHIP.' +
        'RELATING_DRAUGHTING_CALLOUT') | dcr.name = 'suffix')) = 1));
END_ENTITY;


ENTITY styled_item
 SUBTYPE OF (representation_item);
 styles : SET[1:?] OF presentation_style_assignment;
 item : representation_item;
 WHERE
  wr1 : (SIZEOF(SELF.styles) = 1) XOR (SIZEOF(QUERY(pres_style <* SELF.styles
         | NOT ('AUTOMOTIVE_DESIGN.' + 'PRESENTATION_STYLE_BY_CONTEXT' IN
        TYPEOF(pres_style)))) = 0);
END_ENTITY;


ENTITY surface
 SUPERTYPE OF (ONEOF(elementary_surface,swept_surface,bounded_surface,
              offset_surface,surface_replica))
 SUBTYPE OF (geometric_representation_item);
END_ENTITY;


ENTITY surface_condition
 SUBTYPE OF (property_definition);
```

```
WHERE
 wr1 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | NOT (pdr.
       used_representation.description IN ['coating layer','contact ration',
       'hardness','roughness','tactile appearance','treatment result',
       'visual appearance']))) = 0;
 wr2 : SIZEOF(['AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_SHAPE',
       'AUTOMOTIVE_DESIGN.SHAPE_ASPECT'] * TYPEOF(SELF.definition)) = 1;
 wr3 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
       used_representation.desciption = 'coating layer')) = 0) OR (SIZEOF(
       QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
       used_representation.desciption = 'coating layer') AND (SIZEOF(pdr.
       used_representation.items) = 1) AND (SIZEOF(QUERY(r_i <* pdr.
       used_representation.items | (SIZEOF([
       'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(r_i)) = 2) AND
       (r_i.name = 'thickness'))) = 1))) = SIZEOF(QUERY(pdr <* USEDIN(SELF,
       'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
       ) | pdr.used_representation.desciption = 'coating layer')));
 wr4 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
       used_representation.desciption = 'contact ratio')) = 0) OR (SIZEOF(
       QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
       used_representation.desciption = 'contact ratio') AND (SIZEOF(QUERY(
       pdr1 <* USEDIN(pdr.used_representation,
       'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION_' +
       'REPRESENTATION.USED_REPRESENTATION') | (pdr1.name =
       'measuring reference') AND ('AUTOMOTIVE_DESIGN.SURFACE_CONDITION' IN
       TYPEOF(pdr1.definition)) AND (
       'AUTOMOTIVE_DESIGN.PHYSICALLY_MODELLED_PRODUCT_DEFINITION' IN TYPEOF(
       pdr1.definition\property_definition.definition)))) = 1))) > 0);
 wr5 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
       used_representation.desciption = 'contact ratio')) = 0) OR (SIZEOF(
       QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
       used_representation.desciption = 'contact ratio') AND (SIZEOF(pdr.
       used_representation.items) = 1) AND (SIZEOF(QUERY(r_i <* pdr.
       used_representation.items | (SIZEOF([
       'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(r_i)) = 2) AND
       (r_i.name = 'ratio value'))) = 1))) = SIZEOF(QUERY(pdr <* USEDIN(SELF,
       'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
       ) | pdr.used_representation.desciption = 'contact ratio')));
 wr6 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
       used_representation.desciption = 'hardness')) = 0) OR (SIZEOF(QUERY(
       pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
```

```
      used_representation.desciption = 'hardness') AND (
      'AUTOMOTIVE_DESIGN.MATERIAL_PROPERTY_REPRESENTATION' IN TYPEOF(pdr))))
       = SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
      used_representation.desciption = 'hardness')));
wr7 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
      used_representation.desciption = 'hardness')) = 0) OR (SIZEOF(QUERY(
      pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
      used_representation.desciption = 'hardness') AND ({2<=SIZEOF(pdr.
      used_representation.items)<=5}) AND (SIZEOF(QUERY(r_i <* pdr.
      used_representation.items | ((
      'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
      AND (r_i.name IN ['description','measuring method',
      'measuring position'])) OR ((
      'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (
      r_i.name = 'hardness value')) OR ((SIZEOF([
      'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(r_i)) = 2) AND
      (r_i.name = 'depth')))) = SIZEOF(pdr.used_representation.items)))) =
      SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
      used_representation.desciption = 'hardness')));
wr8 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
      used_representation.desciption = 'hardness')) = 0) OR (SIZEOF(QUERY(
      pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
      used_representation.desciption = 'hardness') AND (SIZEOF(QUERY(r_i <*
      pdr.used_representation.items | (
      'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
      AND (r_i.name = 'description'))) = 1))) <= 1);
wr9 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
      used_representation.desciption = 'hardness')) = 0) OR (SIZEOF(QUERY(
      pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
      used_representation.desciption = 'hardness') AND (SIZEOF(QUERY(r_i <*
      pdr.used_representation.items | (
      'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
      AND (r_i.name = 'measuring method'))) = 1))) = 1);
wr10 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
       used_representation.desciption = 'hardness')) = 0) OR (SIZEOF(QUERY(
       pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
       used_representation.desciption = 'hardness') AND (SIZEOF(QUERY(r_i <*
       pdr.used_representation.items | (
       'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
       AND (r_i.name = 'measuring position'))) = 1))) <= 1);
wr11 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
```

```
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'hardness')) = 0) OR (SIZEOF(QUERY(
        pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'hardness') AND (SIZEOF(QUERY(r_i <*
        pdr.used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(r_i)) = 2) AND
         (r_i.name = 'measuring position'))) = 1))) <= 1);
wr12 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'hardness')) = 0) OR (SIZEOF(QUERY(
        pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'hardness') AND (SIZEOF(QUERY(r_i <*
        pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (
        r_i.name = 'hardness value'))) = 1))) = 1);
wr13 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'roughness')) = 0) OR (SIZEOF(QUERY(
        pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'roughness') AND ({2<=SIZEOF(pdr.
        used_representation.items)<=6}) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | ((
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
        AND (r_i.name = 'measuring method')) OR ((
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (
        r_i.name IN ['roughness max value','roughness min value'])) OR ((
        SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(r_i)) = 2) AND
         (r_i.name IN ['cutoff value','sampling length'])) OR ((SIZEOF([
        'AUTOMOTIVE_DESIGN.PCURVE','AUTOMOTIVE_DESIGN.SURFACE_CURVE'] *
        TYPEOF(r_i)) = 1) AND (r_i.name = 'measuring direction')))) = SIZEOF(
        pdr.used_representation.items)))) = SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'roughness')));
wr14 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'roughness')) = 0) OR (SIZEOF(QUERY(
        pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'roughness') AND (SIZEOF(QUERY(r_i <*
        pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (
        r_i.name = 'roughness max value'))) = 1))) = 1);
wr15 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'roughness')) = 0) OR (SIZEOF(QUERY(
        pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
```

```
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'roughness') AND (SIZEOF(QUERY(r_i <*
        pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(r_i)) AND (
        r_i.name = 'roughness min value'))) = 1))) <= 1);
wr16 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'roughness')) = 0) OR (SIZEOF(QUERY(
        pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'roughness') AND (SIZEOF(QUERY(r_i <*
        pdr.used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(r_i)) = 2) AND
         (r_i.name = 'cutoff value'))) = 1))) <= 1);
wr17 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'roughness')) = 0) OR (SIZEOF(QUERY(
        pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'roughness') AND (SIZEOF(QUERY(r_i <*
        pdr.used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(r_i)) = 2) AND
         (r_i.name = 'sampling length'))) = 1))) <= 1);
wr18 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'roughness')) = 0) OR (SIZEOF(QUERY(
        pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'roughness') AND (SIZEOF(QUERY(r_i <*
        pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
        AND (r_i.name = 'measuring method'))) = 1))) = 1);
wr19 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'roughness')) = 0) OR (SIZEOF(QUERY(
        pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'roughness') AND (SIZEOF(QUERY(r_i <*
        pdr.used_representation.items | (SIZEOF(['AUTOMOTIVE_DESIGN.PCURVE',
        'AUTOMOTIVE_DESIGN.SURFACE_CURVE'] * TYPEOF(r_i)) = 1) AND (r_i.name
        = 'measuring method'))) = 1))) = 1);
wr20 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'tactile appearance')) = 0) OR (
        SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'tactile appearance') AND ({1<=
        SIZEOF(pdr.used_representation.items)<=2}) AND (SIZEOF(QUERY(r_i <*
        pdr.used_representation.items | ((
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
```

```
        AND (r_i.name = 'description')) OR ((SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(r_i)) = 2) AND
         (r_i.name = 'depth')))) = SIZEOF(pdr.used_representation.items)))) =
         SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'tactile appearance')));
wr21 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'tactile appearance')) = 0) OR (
        SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'tactile appearance') AND (SIZEOF(
        QUERY(r_i <* pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
        AND (r_i.name = 'description'))) = 1))) <= 1);
wr22 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'tactile appearance')) = 0) OR (
        SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'tactile appearance') AND (SIZEOF(
        QUERY(r_i <* pdr.used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.PCURVE','AUTOMOTIVE_DESIGN.SURFACE_CURVE'] *
        TYPEOF(r_i)) = 1) AND (r_i.name = 'depth'))) = 1))) <= 1);
wr23 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'treatmen result')) = 0) OR (SIZEOF(
        QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'treatmen result') AND ({1<=SIZEOF(
        pdr.used_representation.items)<=2}) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | ((
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
        AND (r_i.name IN ['description','purpose']))))) = SIZEOF(pdr.
        used_representation.items)))) = SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'treatmen result')));
wr24 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'treatmen result')) = 0) OR (SIZEOF(
        QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'treatmen result') AND (SIZEOF(QUERY
        (r_i <* pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
        AND (r_i.name = 'description'))) = 1))) = 1);
wr25 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'treatmen result')) = 0) OR (SIZEOF(
        QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
```

```
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'treatmen result') AND (SIZEOF(QUERY
        (r_i <* pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
        AND (r_i.name = 'purpose'))) = 1))) <= 1);
wr26 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'visual appearance')) = 0) OR (
        SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'visual appearance') AND ({2<=SIZEOF
        (pdr.used_representation.items)<=5}) AND (SIZEOF(QUERY(r_i <* pdr.
        used_representation.items | ((
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
        AND (r_i.name IN ['colour id','colour name','lustre','pattern',
        'transparency']))))) = SIZEOF(pdr.used_representation.items)))) =
        SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'visual appearance')));
wr27 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'visual appearance')) = 0) OR (
        SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'visual appearance') AND (SIZEOF(
        QUERY(r_i <* pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
        AND (r_i.name = 'colour id'))) = 1))) = 1);
wr28 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'visual appearance')) = 0) OR (
        SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'visual appearance') AND (SIZEOF(
        QUERY(r_i <* pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
        AND (r_i.name = 'colour name'))) = 1))) <= 1);
wr29 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'visual appearance')) = 0) OR (
        SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'visual appearance') AND (SIZEOF(
        QUERY(r_i <* pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
        AND (r_i.name = 'lustre'))) = 1))) = 1);
wr30 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'visual appearance')) = 0) OR (
        SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'visual appearance') AND (SIZEOF(
```

```
        QUERY(r_i <* pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
        AND (r_i.name = 'pattern'))) = 1))) <= 1);
  wr31 : (SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | pdr.
        used_representation.desciption = 'visual appearance')) = 0) OR (
        SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (pdr.
        used_representation.desciption = 'visual appearance') AND (SIZEOF(
        QUERY(r_i <* pdr.used_representation.items | (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(r_i))
        AND (r_i.name = 'transparency'))) = 1))) <= 1);
END_ENTITY;


ENTITY surface_condition_callout
 SUBTYPE OF (draughting_callout);
END_ENTITY;


ENTITY surface_curve
 SUPERTYPE OF (ONEOF(intersection_curve,seam_curve))
 SUBTYPE OF (curve);
 curve_3d : curve;
 associated_geometry : LIST[1:2] OF pcurve_or_surface;
 master_representation : preferred_surface_curve_representation;
 DERIVE
  basis_surface : SET[1:2] OF surface := get_basis_surface(SELF);
 WHERE
  wr1 : curve_3d.dim = 3;
  wr2 : ('AUTOMOTIVE_DESIGN.PCURVE' IN TYPEOF(associated_geometry[1])) OR (
        master_representation <> pcurve_s1);
  wr3 : ('AUTOMOTIVE_DESIGN.PCURVE' IN TYPEOF(associated_geometry[2])) OR (
        master_representation <> pcurve_s2);
  wr4 : NOT ('AUTOMOTIVE_DESIGN.PCURVE' IN TYPEOF(curve_3d));
END_ENTITY;


ENTITY surface_of_linear_extrusion
 SUBTYPE OF (swept_surface);
 extrusion_axis : vector;
END_ENTITY;


ENTITY surface_of_revolution
 SUBTYPE OF (swept_surface);
 axis_position : axis1_placement;
 DERIVE
  axis_line : line := line(axis_position.location,vector(axis_position.z,1.0))
             ;
END_ENTITY;


ENTITY surface_pair
```

```
 SUBTYPE OF (kinematic_pair);
 surface_1 : surface;
 surface_2 : surface;
 orientation : BOOLEAN;
 WHERE
  wr1 : frame_associated_to_background(SELF\kinematic_pair.
        pair_placement_in_first_link_context,surface_1);
  wr2 : frame_associated_to_background(SELF\kinematic_pair.
        pair_placement_in_second_link_context,surface_2);
END_ENTITY;


ENTITY surface_pair_range
 SUBTYPE OF (simple_pair_range);
 SELF\simple_pair_range.applies_to_pair : surface_pair;
 range_on_surface_1 : rectangular_trimmed_surface;
 range_on_surface_2 : rectangular_trimmed_surface;
 lower_limit_actual_rotation : rotational_range_measure;
 upper_limit_actual_rotation : rotational_range_measure;
 WHERE
  wr1 : SELF\simple_pair_range.applies_to_pair\surface_pair.surface_1 :=:
        range_on_surface_1.basis_surface;
  wr2 : SELF\simple_pair_range.applies_to_pair\surface_pair.surface_2 :=:
        range_on_surface_2.basis_surface;
  wr3 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        lower_limit_actual_rotation)) OR ('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE'
        IN TYPEOF(upper_limit_actual_rotation))) XOR (
        lower_limit_actual_rotation < upper_limit_actual_rotation);
END_ENTITY;


ENTITY surface_patch;
 parent_surface : bounded_surface;
 u_transition : transition_code;
 v_transition : transition_code;
 u_sense : BOOLEAN;
 v_sense : BOOLEAN;
 INVERSE
  using_surfaces : BAG[1:?] OF rectangular_composite_surface FOR segments;
 WHERE
  wr1 : (NOT ('AUTOMOTIVE_DESIGN.CURVE_BOUNDED_SURFACE' IN TYPEOF(
        parent_surface)));
END_ENTITY;


ENTITY surface_profile_tolerance
 SUBTYPE OF (geometric_tolerance);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT' IN
         TYPEOF(SELF));
  wr2 : (NOT (('AUTOMOTIVE_DESIGN.' +
        'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF))) OR (
        SIZEOF(SELF\geometric_tolerance_with_datum_reference.datum_system) <=
```

```
        3);
END_ENTITY;


ENTITY surface_rendering_properties;
 rendered_colour : colour;
END_ENTITY;


ENTITY surface_replica
 SUBTYPE OF (surface);
 parent_surface : surface;
 transformation : cartesian_transformation_operator_3d;
 WHERE
  wr1 : acyclic_surface_replica(SELF,parent_surface);
END_ENTITY;


ENTITY surface_side_style;
 name : label;
 styles : SET[1:7] OF surface_style_element_select;
 WHERE
  wr1 : SIZEOF(QUERY(style1 <* SELF.styles | SIZEOF(QUERY(style2 <* SELF.
        styles - style1 | TYPEOF(style1) = TYPEOF(style2))) > 0)) = 0;
END_ENTITY;


ENTITY surface_style_boundary;
 style_of_boundary : curve_or_render;
END_ENTITY;


ENTITY surface_style_control_grid;
 style_of_control_grid : curve_or_render;
END_ENTITY;


ENTITY surface_style_fill_area;
 fill_area : fill_area_style;
END_ENTITY;


ENTITY surface_style_parameter_line;
 style_of_parameter_lines : curve_or_render;
 direction_counts : SET[1:2] OF direction_count_select;
 WHERE
  wr1 : (HIINDEX(SELF.direction_counts) = 1) XOR (TYPEOF(SELF.direction_counts
        [1]) <> TYPEOF(SELF.direction_counts[2]));
END_ENTITY;


ENTITY surface_style_reflectance_ambient;
 ambient_reflectance : REAL;
END_ENTITY;
```

```
ENTITY surface_style_reflectance_ambient_diffuse
 SUBTYPE OF (surface_style_reflectance_ambient);
 diffuse_reflectance : REAL;
END_ENTITY;


ENTITY surface_style_reflectance_ambient_diffuse_specular
 SUBTYPE OF (surface_style_reflectance_ambient_diffuse);
 specular_reflectance : REAL;
 specular_exponent : REAL;
 specular_colour : colour;
END_ENTITY;


ENTITY surface_style_rendering;
 rendering_method : shading_surface_method;
 surface_colour : colour;
END_ENTITY;


ENTITY surface_style_rendering_with_properties
 SUBTYPE OF (surface_style_rendering);
 properties : SET[1:2] OF rendering_properties_select;
 WHERE
  wr1 : (HIINDEX(SELF.properties) = 1) XOR (TYPEOF(SELF.properties[1]) <>
        TYPEOF(SELF.properties[2]));
END_ENTITY;


ENTITY surface_style_segmentation_curve;
 style_of_segmentation_curve : curve_or_render;
END_ENTITY;


ENTITY surface_style_silhouette;
 style_of_silhouette : curve_or_render;
END_ENTITY;


ENTITY surface_style_transparent;
 transparency : REAL;
 WHERE
  wr1 : {0.0<=transparency<=1.0};
END_ENTITY;


ENTITY surface_style_usage;
 side : surface_side;
 style : surface_side_style_select;
END_ENTITY;


ENTITY swept_face_solid
 SUPERTYPE OF (ONEOF(extruded_face_solid,revolved_face_solid))
 SUBTYPE OF (solid_model);
```

```
 swept_face : face_surface;
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.PLANE' IN TYPEOF(swept_face.face_geometry);
END_ENTITY;


ENTITY swept_surface
 SUPERTYPE OF (ONEOF(surface_of_linear_extrusion,surface_of_revolution))
 SUBTYPE OF (surface);
 swept_curve : curve;
END_ENTITY;


ENTITY symbol_colour;
 colour_of_symbol : colour;
END_ENTITY;


ENTITY symbol_representation
 SUBTYPE OF (representation);
END_ENTITY;


ENTITY symbol_representation_map
 SUBTYPE OF (representation_map);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.SYMBOL_REPRESENTATION' IN TYPEOF(SELF\
        representation_map.mapped_representation);
  wr2 : 'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT' IN TYPEOF(SELF\representation_map.
        mapping_origin);
END_ENTITY;


ENTITY symbol_style;
 name : label;
 style_of_symbol : symbol_style_select;
END_ENTITY;


ENTITY symbol_target
 SUBTYPE OF (geometric_representation_item);
 placement : axis2_placement;
 x_scale : positive_ratio_measure;
 y_scale : positive_ratio_measure;
END_ENTITY;


ENTITY symmetric_shape_aspect
 SUBTYPE OF (shape_aspect);
 INVERSE
  basis_relationships : SET[1:?] OF shape_aspect_relationship FOR
                        relating_shape_aspect;
 WHERE
  wr1 : SIZEOF(QUERY(x <* SELF.basis_relationships |
        'AUTOMOTIVE_DESIGN.CENTRE_OF_SYMMETRY' IN TYPEOF(x.
```

```
        related_shape_aspect))) >= 1;
END_ENTITY;


ENTITY symmetry_tolerance
 SUBTYPE OF (geometric_tolerance_with_datum_reference);
 WHERE
  wr1 : SIZEOF(SELF\geometric_tolerance_with_datum_reference.datum_system) = 1
        ;
  wr2 : NOT ('AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT' IN
         TYPEOF(SELF));
END_ENTITY;


ENTITY tangent
 SUBTYPE OF (derived_shape_aspect);
 WHERE
  wr1 : SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
END_ENTITY;


ENTITY taper
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SELF.description IN ['angle taper','diameter taper','directed taper'];
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(pdr.used_representation.items) = 1)
        )) = 1;
  wr5 : (SELF.description <> 'angle taper') OR (SIZEOF(QUERY(pdr <* USEDIN(
        SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) =
         2) AND (srwp_i.name = 'taper angle'))) = 1))) = 1);
  wr6 : (SELF.description <> 'diameter taper') OR (SIZEOF(QUERY(pdr <* USEDIN(
        SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
```

```
        AND (srwp_i.name = 'final diameter'))) = 1))) = 1);
  wr7 : (SELF.description <> 'directed taper') OR (SIZEOF(QUERY(pdr <* USEDIN(
        SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) =
         2) AND (srwp_i.name = 'draft angle'))) = 1))) = 1);
  wr8 : (SELF.description <> 'directed taper') OR (SIZEOF(QUERY(pdr <* USEDIN(
        SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation)))) = 1) AND (SIZEOF(QUERY(pdr <* USEDIN(SELF,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation)) AND (pdr.used_representation.description =
        'draft direction'))) = 1);
END_ENTITY;


ENTITY tee_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1;
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND ({9<=SIZEOF(pdr.used_representation.items)
        <=10}))) = 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | NOT (srwp_i.name IN ['orientation','width'
        ,'depth','cross bar width','cross bar depth','first offset',
        'second offset','first angle','second angle','radius']))) > 0))) = 0;
  wr5 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'orientation') AND (
        'AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i)))) = 1))) = 1;
  wr6 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
```

```
      used_representation.items | (srwp_i.name = 'width') AND (SIZEOF([
      'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
       = 1))) = 1;
wr7 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
      .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (srwp_i.name = 'depth') AND (SIZEOF([
      'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
       = 1))) = 1;
wr8 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
      .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (srwp_i.name = 'cross bar width') AND (
      SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
       = 1))) = 1;
wr9 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
      'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
      .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (srwp_i.name = 'cross bar depth') AND (
      SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)))
       = 1))) = 1;
wr10 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
       'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
       pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
       used_representation.items | (srwp_i.name = 'first offset') AND (
       SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2))
       ) = 1))) = 1;
wr11 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
       'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
       pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
       used_representation.items | (srwp_i.name = 'second offset') AND (
       SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2))
       ) = 1))) = 1;
wr12 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
       'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
       pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
       used_representation.items | (srwp_i.name = 'first angle') AND (SIZEOF
       (['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
       'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i))
       = 2))) = 1))) = 1;
```

```
  wr13 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
         'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
         pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
         used_representation.items | (srwp_i.name = 'second angle') AND (
         SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i))
         = 2))) = 1))) = 1;
  wr14 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
         'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
         pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
         used_representation.items | (srwp_i.name = 'radius') AND (SIZEOF([
         'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
         'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i))
         = 2))) = 1))) <= 1;
END_ENTITY;


ENTITY terminator_symbol
 SUBTYPE OF (annotation_symbol_occurrence);
 annotated_curve : annotation_curve_occurrence;
END_ENTITY;


ENTITY text_literal
 SUBTYPE OF (geometric_representation_item);
 literal : presentable_text;
 placement : axis2_placement;
 alignment : text_alignment;
 path : text_path;
 font : font_select;
END_ENTITY;


ENTITY text_literal_with_associated_curves
 SUBTYPE OF (text_literal);
 associated_curves : SET[1:?] OF curve;
END_ENTITY;


ENTITY text_literal_with_blanking_box
 SUBTYPE OF (text_literal);
 blanking : planar_box;
END_ENTITY;


ENTITY text_literal_with_delineation
 SUBTYPE OF (text_literal);
 delineation : text_delineation;
END_ENTITY;


ENTITY text_literal_with_extent
```

```
 SUBTYPE OF (text_literal);
 extent : planar_extent;
END_ENTITY;

ENTITY text_string_representation
 SUBTYPE OF (representation);
 WHERE
  wr1 : SIZEOF(QUERY(item <* SELF\representation.items | SIZEOF([
        'AUTOMOTIVE_DESIGN.TEXT_LITERAL','AUTOMOTIVE_DESIGN.ANNOTATION_TEXT',
        'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_CHARACTER',
        'AUTOMOTIVE_DESIGN.DEFINED_CHARACTER_GLYPH',
        'AUTOMOTIVE_DESIGN.COMPOSITE_TEXT','AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT'
        ] * TYPEOF(item)) = 0)) = 0;
  wr2 : SIZEOF(QUERY(item <* SELF\representation.items | NOT (SIZEOF([
        'AUTOMOTIVE_DESIGN.TEXT_LITERAL','AUTOMOTIVE_DESIGN.ANNOTATION_TEXT',
        'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT_CHARACTER',
        'AUTOMOTIVE_DESIGN.DEFINED_CHARACTER_GLYPH',
        'AUTOMOTIVE_DESIGN.COMPOSITE_TEXT'] * TYPEOF(item)) = 0))) >= 1;
  wr3 : SIZEOF(QUERY(a2p <* QUERY(item <* SELF\representation.items |
        'AUTOMOTIVE_DESIGN.AXIS2_PLACEMENT' IN TYPEOF(item)) | NOT ((SIZEOF(
        QUERY(at <* QUERY(item <* SELF\representation.items |
        'AUTOMOTIVE_DESIGN.' + 'ANNOTATION_TEXT' IN TYPEOF(item)) | (at\
        mapped_item.mapping_target :=: a2p))) >= 1) OR (SIZEOF(QUERY(atc <*
        QUERY(item <* SELF\representation.items | 'AUTOMOTIVE_DESIGN.' +
        'ANNOTATION_TEXT_CHARACTER' IN TYPEOF(item)) | (atc\mapped_item.
        mapping_target :=: a2p))) >= 1)))) = 0;
END_ENTITY;

ENTITY text_style;
 name : label;
 character_appearance : character_style_select;
END_ENTITY;

ENTITY text_style_for_defined_font;
 text_colour : colour;
END_ENTITY;

ENTITY text_style_with_box_characteristics
 SUBTYPE OF (text_style);
 characteristics : SET[1:4] OF box_characteristic_select;
 WHERE
  wr1 : SIZEOF(QUERY(c1 <* SELF.characteristics | SIZEOF(QUERY(c2 <* SELF.
        characteristics - c1 | TYPEOF(c1) = TYPEOF(c2))) > 0)) = 0;
END_ENTITY;

ENTITY text_style_with_mirror
 SUBTYPE OF (text_style);
 mirror_placement : axis2_placement;
END_ENTITY;
```

```
ENTITY text_style_with_spacing
 SUBTYPE OF (text_style);
 character_spacing : character_spacing_select;
END_ENTITY;


ENTITY thermodynamic_temperature_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.THERMODYNAMIC_TEMPERATURE_UNIT' IN TYPEOF(SELF\
        measure_with_unit.unit_component);
END_ENTITY;


ENTITY thermodynamic_temperature_unit
 SUBTYPE OF (named_unit);
 WHERE
  wr1 : (SELF\named_unit.dimensions.length_exponent = 0.0) AND (SELF\
        named_unit.dimensions.mass_exponent = 0.0) AND (SELF\named_unit.
        dimensions.time_exponent = 0.0) AND (SELF\named_unit.dimensions.
        electric_current_exponent = 0.0) AND (SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 1.0) AND (SELF\named_unit.
        dimensions.amount_of_substance_exponent = 0.0) AND (SELF\named_unit.
        dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;


ENTITY thread
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.used_representation.
        items) = 10))) = 1)) = 1;
  wr2 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (
        'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i))
         AND (srwp_i.name = 'qualifier'))) = 1))) = 1)) = 1;
  wr3 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
        'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
        ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'major diameter'))) = 1))) = 1)) = 1;
```

```
wr4 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (SIZEOF([
      'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
      AND (srwp_i.name = 'minor diameter'))) = 1))) = 1)) = 1;
wr5 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (SIZEOF([
      'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
      AND (srwp_i.name = 'pitch diameter'))) = 1))) = 1)) = 1;
wr6 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (SIZEOF([
      'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
      AND (srwp_i.name = 'crest'))) = 1))) = 1)) = 1;
wr7 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (SIZEOF([
      'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
      'AUTOMOTIVE_DESIGN.RATIO_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
      AND (srwp_i.name = 'density'))) = 1))) = 1)) = 1;
wr8 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (
      'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i))
       AND (srwp_i.name = 'fit class'))) = 1))) = 1)) = 1;
wr9 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
      'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
      'AUTOMOTIVE_DESIGN.' + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'
      ) | ('AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
      used_representation.items | (
      'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i))
       AND (srwp_i.name = 'form'))) = 1))) = 1)) = 1;
```

```
  wr10 : SIZEOF(QUERY(pd <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION.DEFINITION') | SIZEOF(QUERY(pdr <* USEDIN(pd,
         'AUTOMOTIVE_DESIGN.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
         'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(
         pdr.used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
         used_representation.items | (
         'AUTOMOTIVE_DESIGN.DESCRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(srwp_i)
         ) AND (srwp_i.name = 'hand'))) = 1))) = 1)) = 1;
END_ENTITY;


ENTITY thread_occurrence
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : SIZEOF(['AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION',
        'AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_SOLID'] * TYPEOF(SELF.of_shape.
        definition)) = 1;
  wr2 : SELF.description IN ['inner','outer'];
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND ({2<=SIZEOF(pdr.used_representation.items)
        <=3}))) = 1;
  wr5 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | NOT (srwp_i.name IN ['orientation',
        'effective length','maximum length']))) > 0))) = 0;
  wr6 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'effective length'))) = 1))) = 1;
  wr7 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | ('AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(
        srwp_i)) AND (srwp_i.name = 'orientation'))) = 1))) = 1;
  wr8 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (SIZEOF([
```

```
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) = 2)
        AND (srwp_i.name = 'maximum length'))) <= 1))) = 1;
  wr9 : SIZEOF(QUERY(sar <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') | SIZEOF([
        'AUTOMOTIVE_DESIGN.THREAD',
        'AUTOMOTIVE_DESIGN.EXTERNALLY_DEFINED_THREAD'] * TYPEOF(sar.
        relating_shape_aspect.of_shape.definition)) = 1)) = 1;
END_ENTITY;


ENTITY time_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.TIME_UNIT' IN TYPEOF(SELF\measure_with_unit.
        unit_component);
END_ENTITY;


ENTITY time_unit
 SUBTYPE OF (named_unit);
 WHERE
  wr1 : (SELF\named_unit.dimensions.length_exponent = 0.0) AND (SELF\
        named_unit.dimensions.mass_exponent = 0.0) AND (SELF\named_unit.
        dimensions.time_exponent = 1.0) AND (SELF\named_unit.dimensions.
        electric_current_exponent = 0.0) AND (SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 0.0) AND (SELF\named_unit.
        dimensions.amount_of_substance_exponent = 0.0) AND (SELF\named_unit.
        dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;


ENTITY tolerance_value;
 lower_bound : measure_with_unit;
 upper_bound : measure_with_unit;
 WHERE
  wr1 : upper_bound.value_component > lower_bound.value_component;
  wr2 : upper_bound.unit_component = lower_bound.unit_component;
END_ENTITY;


ENTITY tolerance_zone
 SUBTYPE OF (shape_aspect);
 defining_tolerance : SET[1:?] OF geometric_tolerance;
 form : tolerance_zone_form;
END_ENTITY;


ENTITY tolerance_zone_definition
 SUPERTYPE OF (ONEOF(projected_zone_definition,runout_zone_definition));
 zone : tolerance_zone;
 boundaries : SET[1:?] OF shape_aspect;
END_ENTITY;
```

```
ENTITY tolerance_zone_form;
 name : label;
END_ENTITY;


ENTITY topological_representation_item
 SUPERTYPE OF (ONEOF(vertex,edge,face_bound,face,connected_edge_set,
               connected_face_set,(loop ANDOR path)))
 SUBTYPE OF (representation_item);
END_ENTITY;


ENTITY toroidal_surface
 SUBTYPE OF (elementary_surface);
 major_radius : positive_length_measure;
 minor_radius : positive_length_measure;
END_ENTITY;


ENTITY torus
 SUBTYPE OF (geometric_representation_item);
 position : axis1_placement;
 major_radius : positive_length_measure;
 minor_radius : positive_length_measure;
 WHERE
  wr1 : major_radius > minor_radius;
END_ENTITY;


ENTITY total_runout_tolerance
 SUBTYPE OF (geometric_tolerance_with_datum_reference);
 WHERE
  wr1 : NOT ('AUTOMOTIVE_DESIGN.' + 'GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT' IN
         TYPEOF(SELF));
  wr2 : SIZEOF(SELF\geometric_tolerance_with_datum_reference.datum_system) <=
        2;
END_ENTITY;


ENTITY transition_feature
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : SIZEOF(['AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION',
        'AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_PANEL',
        'AUTOMOTIVE_DESIGN.COMPOUND_FEATURE_IN_SOLID'] * TYPEOF(SELF.of_shape.
        definition)) = 1;
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation)) AND (pdr.used_representation.description =
        'first face shape'))) = 1;
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
```

```
        used_representation)) AND (pdr.used_representation.description =
        'second face shape'))) = 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation)) AND (pdr.used_representation.description =
        'transition face shape'))) = 1;
  wr5 : SIZEOF(['AUTOMOTIVE_DESIGN.CHAMFER','AUTOMOTIVE_DESIGN.EDGE_ROUND',
        'AUTOMOTIVE_DESIGN.FILLET'] * TYPEOF(SELF)) = 1;
END_ENTITY;


ENTITY trimmed_curve
 SUBTYPE OF (bounded_curve);
 basis_curve : curve;
 trim_1 : SET[1:2] OF trimming_select;
 trim_2 : SET[1:2] OF trimming_select;
 sense_agreement : BOOLEAN;
 master_representation : trimming_preference;
 WHERE
  wr1 : (HIINDEX(trim_1) = 1) XOR (TYPEOF(trim_1[1]) <> TYPEOF(trim_1[2]));
  wr2 : (HIINDEX(trim_2) = 1) XOR (TYPEOF(trim_2[1]) <> TYPEOF(trim_2[2]));
END_ENTITY;


ENTITY two_direction_repeat_factor
 SUBTYPE OF (one_direction_repeat_factor);
 second_repeat_factor : vector;
END_ENTITY;


ENTITY type_qualifier;
 name : label;
END_ENTITY;


ENTITY uncertainty_assigned_representation
 SUBTYPE OF (representation);
 uncertainty : SET[1:?] OF uncertainty_measure_with_unit;
END_ENTITY;


ENTITY uncertainty_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 name : label;
 description : text;
 WHERE
  wr1 : (NOT ('NUMBER' IN TYPEOF(SELF\measure_with_unit.value_component))) OR
        (SELF\measure_with_unit.value_component >= 0);
END_ENTITY;


ENTITY uncertainty_qualifier
 SUPERTYPE OF (ONEOF(standard_uncertainty,qualitative_uncertainty));
```

```
 measure_name : label;
 description : text;
END_ENTITY;


ENTITY unconstrained_pair
 SUBTYPE OF (kinematic_pair);
END_ENTITY;


ENTITY unconstrained_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : unconstrained_pair;
 actual_placement : axis2_placement_3d;
END_ENTITY;


ENTITY uniform_curve
 SUBTYPE OF (b_spline_curve);
END_ENTITY;


ENTITY uniform_surface
 SUBTYPE OF (b_spline_surface);
END_ENTITY;


ENTITY universal_pair
 SUBTYPE OF (kinematic_pair);
 input_skew_angle : OPTIONAL plane_angle_measure;
 DERIVE
  skew_angle : plane_angle_measure := NVL(input_skew_angle,0.0);
 WHERE
  wr1 : COS(plane_angle_for_pair_in_radian(SELF\kinematic_pair,skew_angle)) >
        0.0;
END_ENTITY;


ENTITY universal_pair_range
 SUBTYPE OF (simple_pair_range);
 SELF\simple_pair_range.applies_to_pair : universal_pair;
 lower_limit_first_rotation : rotational_range_measure;
 upper_limit_first_rotation : rotational_range_measure;
 lower_limit_second_rotation : rotational_range_measure;
 upper_limit_second_rotation : rotational_range_measure;
 WHERE
  wr1 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        lower_limit_first_rotation)) OR ('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE'
        IN TYPEOF(upper_limit_first_rotation))) XOR (
        lower_limit_first_rotation < upper_limit_first_rotation);
  wr2 : (('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE' IN TYPEOF(
        lower_limit_second_rotation)) OR ('AUTOMOTIVE_DESIGN.UNLIMITED_RANGE'
        IN TYPEOF(upper_limit_second_rotation))) XOR (
        lower_limit_second_rotation < upper_limit_second_rotation);
```

```
END_ENTITY;


ENTITY universal_pair_value
 SUBTYPE OF (pair_value);
 SELF\pair_value.applies_to_pair : universal_pair;
 first_rotation_angle : plane_angle_measure;
 second_rotation_angle : plane_angle_measure;
END_ENTITY;


ENTITY vector
 SUBTYPE OF (geometric_representation_item);
 orientation : direction;
 magnitude : length_measure;
 WHERE
  wr1 : magnitude >= 0.0;
END_ENTITY;


ENTITY vector_style
 SUBTYPE OF (curve_style,pre_defined_terminator_symbol);
END_ENTITY;


ENTITY vee_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF.
        of_shape.definition);
  wr2 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation))) = 1;
  wr3 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND ({3<=SIZEOF(pdr.used_representation.items)
        <=4}))) = 1;
  wr4 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | NOT (srwp_i.name IN ['orientation',
        'profile angle','tilt angle','profile radius']))) > 0))) = 0;
  wr5 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'orientation') AND (
        'AUTOMOTIVE_DESIGN.PLACEMENT' IN TYPEOF(srwp_i)))) = 1))) = 1;
  wr6 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
```

```
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'profile angle') AND (
        SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) =
         2))) = 1))) = 1;
  wr7 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'tilt angle') AND (SIZEOF([
        'AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) =
         2))) = 1))) = 1;
  wr8 : SIZEOF(QUERY(pdr <* USEDIN(SELF,'AUTOMOTIVE_DESIGN.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN TYPEOF(pdr
        .used_representation)) AND (SIZEOF(QUERY(srwp_i <* pdr.
        used_representation.items | (srwp_i.name = 'profile radius') AND (
        SIZEOF(['AUTOMOTIVE_DESIGN.MEASURE_REPRESENTATION_ITEM',
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(srwp_i)) =
         2))) = 1))) <= 1;
END_ENTITY;


ENTITY version_assignment
 SUBTYPE OF (identification_assignment);
 items : SET[1:?] OF version_item;
 WHERE
  wr1 : SELF.role.name = 'version id';
  wr2 : SIZEOF(SELF.items) = 1;
END_ENTITY;


ENTITY versioned_action_request;
 id : identifier;
 version : label;
 purpose : text;
 description : text;
END_ENTITY;


ENTITY vertex
 SUBTYPE OF (topological_representation_item);
END_ENTITY;


ENTITY vertex_loop
 SUBTYPE OF (loop);
 loop_vertex : vertex;
END_ENTITY;


ENTITY vertex_point
 SUBTYPE OF (vertex,geometric_representation_item);
```

```
 vertex_geometry : point;
END_ENTITY;


ENTITY view_volume;
 projection_type : central_or_parallel;
 projection_point : cartesian_point;
 view_plane_distance : length_measure;
 front_plane_distance : length_measure;
 front_plane_clipping : BOOLEAN;
 back_plane_distance : length_measure;
 back_plane_clipping : BOOLEAN;
 view_volume_sides_clipping : BOOLEAN;
 view_window : planar_box;
END_ENTITY;


ENTITY volume_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1 : 'AUTOMOTIVE_DESIGN.VOLUME_UNIT' IN TYPEOF(SELF\measure_with_unit.
        unit_component);
END_ENTITY;


ENTITY volume_unit
 SUBTYPE OF (named_unit);
 WHERE
  wr1 : (SELF\named_unit.dimensions.length_exponent = 3.0) AND (SELF\
        named_unit.dimensions.mass_exponent = 0.0) AND (SELF\named_unit.
        dimensions.time_exponent = 0.0) AND (SELF\named_unit.dimensions.
        electric_current_exponent = 0.0) AND (SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 0.0) AND (SELF\named_unit.
        dimensions.amount_of_substance_exponent = 0.0) AND (SELF\named_unit.
        dimensions.luminous_intensity_exponent = 0.0);
END_ENTITY;


FUNCTION acyclic_composite_text(start_composite : composite_text;child_text :
                                SET[1:?] OF text_or_character) : LOGICAL;
 LOCAL
  i : INTEGER;
  local_composite_text : SET[0:?] OF composite_text;
  local_annotation_text : SET[0:?] OF annotation_text;
  local_children : SET[0:?] OF text_or_character;
 END_LOCAL;
 local_composite_text := QUERY(child <* child_text | (
                         'AUTOMOTIVE_DESIGN.COMPOSITE_TEXT' IN TYPEOF(child)))
                         ;
 IF (SIZEOF(local_composite_text) > 0)
  THEN
   REPEAT i := 1 TO HIINDEX(local_composite_text);
    IF (start_composite :=: local_composite_text[i])
     THEN
```

```
      RETURN (FALSE);
    END_IF;
   END_REPEAT;
 END_IF;
 local_children := child_text;
 IF (SIZEOF(local_composite_text)) > 0
  THEN
   REPEAT i := 1 TO HIINDEX(local_composite_text);
    local_children := local_children + local_composite_text[i].collected_text;
   END_REPEAT;
 END_IF;
 local_annotation_text := QUERY(child <* child_text | (
                          'AUTOMOTIVE_DESIGN.ANNOTATION_TEXT' IN TYPEOF(child)
                          ));
 IF (SIZEOF(local_annotation_text) > 0)
  THEN
   REPEAT i := 1 TO HIINDEX(local_annotation_text);
    local_children := local_children + QUERY(item <* local_annotation_text[i]\
                      mapped_item.mapping_source.mapped_representation.items
                       | SIZEOF(['AUTOMOTIVE_DESIGN.ANNOTATION_TEXT',
                      'AUTOMOTIVE_DESIGN.COMPOSITE_TEXT'] * TYPEOF(item)) > 0)
                      ;
   END_REPEAT;
 END_IF;
 IF (local_children :<>: child_text)
  THEN
   RETURN (acyclic_composite_text(start_composite,local_children));
  ELSE
   RETURN (TRUE);
 END_IF;
END_FUNCTION;


FUNCTION acyclic_curve_replica(rep : curve_replica;parent : curve) : BOOLEAN;
 IF NOT (('AUTOMOTIVE_DESIGN.CURVE_REPLICA') IN TYPEOF(parent))
  THEN
   RETURN (TRUE);
 END_IF;
 IF (parent :=: rep)
  THEN
   RETURN (FALSE);
  ELSE
   RETURN (acyclic_curve_replica(rep,parent\curve_replica.parent_curve));
 END_IF;
END_FUNCTION;


FUNCTION acyclic_mapped_item_usage(rep : representation) : BOOLEAN;
 LOCAL
  items : SET OF representation_item;
  i : INTEGER;
 END_LOCAL;
 items := QUERY(item <* rep.items | 'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(
```

```
          item));
  IF SIZEOF(items) = 0
   THEN
    RETURN (FALSE);
   ELSE
    REPEAT i := 1 TO HIINDEX(items);
     IF items[i]\mapped_item.mapping_source.mapped_representation :=: rep
      THEN
       RETURN (TRUE);
      ELSE
       RETURN (acyclic_mapped_item_usage(items[i]\mapped_item.mapping_source.
        mapped_representation));
     END_IF;
    END_REPEAT;
    RETURN (FALSE);
  END_IF;
 END_FUNCTION;


 FUNCTION acyclic_mapped_representation(parent_set : SET OF representation;
                                        children_set : SET OF
                                        representation_item) : BOOLEAN;
  LOCAL
   x,y : SET OF representation_item;
   i : INTEGER;
  END_LOCAL;
  x := QUERY(z <* children_set | 'AUTOMOTIVE_DESIGN.MAPPED_ITEM' IN TYPEOF(z));
  IF SIZEOF(x) > 0
   THEN
    REPEAT i := 1 TO HIINDEX(x);
     IF x[i]\mapped_item.mapping_source.mapped_representation IN parent_set
      THEN
       RETURN (FALSE);
     END_IF;
     IF NOT acyclic_mapped_representation(parent_set + x[i]\mapped_item.
        mapping_source.mapped_representation,x[i]\mapped_item.mapping_source.
        mapped_representation.items)
      THEN
       RETURN (FALSE);
     END_IF;
    END_REPEAT;
  END_IF;
  x := children_set - x;
  IF SIZEOF(x) > 0
   THEN
    REPEAT i := 1 TO HIINDEX(x);
     y := QUERY(z <* bag_to_set(USEDIN(x[i],'')) |
          'AUTOMOTIVE_DESIGN.REPRESENTATION_ITEM' IN TYPEOF(z));
     IF NOT acyclic_mapped_representation(parent_set,y)
      THEN
       RETURN (FALSE);
     END_IF;
    END_REPEAT;
```

```
 END_IF;
 RETURN (TRUE);
END_FUNCTION;


FUNCTION acyclic_point_replica(rep : point_replica;parent : point) : BOOLEAN;
 IF NOT (('AUTOMOTIVE_DESIGN.POINT_REPLICA') IN TYPEOF(parent))
  THEN
   RETURN (TRUE);
 END_IF;
 IF (parent :=: rep)
  THEN
   RETURN (FALSE);
  ELSE
   RETURN (acyclic_point_replica(rep,parent\point_replica.parent_pt));
 END_IF;
END_FUNCTION;


FUNCTION acyclic_product_category_relationship(relation :
                                              product_category_relationship;
                                              children : SET OF
                                              product_category) : LOGICAL;
 LOCAL
  x : SET OF product_category_relationship;
  i : INTEGER;
  local_children : SET OF product_category;
 END_LOCAL;
 REPEAT i := 1 TO HIINDEX(children);
  IF relation.category :=: children[i]
   THEN
    RETURN (FALSE);
  END_IF;
 END_REPEAT;
 x := bag_to_set(USEDIN(relation.category,'AUTOMOTIVE_DESIGN.' +
      'PRODUCT_CATEGORY_RELATIONSHIP.SUB_CATEGORY'));
 local_children := children + relation.category;
 IF SIZEOF(x) > 0
  THEN
   REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_product_category_relationship(x[i],local_children)
     THEN
      RETURN (FALSE);
    END_IF;
   END_REPEAT;
 END_IF;
 RETURN (TRUE);
END_FUNCTION;


FUNCTION acyclic_product_definition_relationship(relation :
         product_definition_relationship;relatives : SET OF product_definition
                                                ;specific_relation : STRING) :
```

```
                                                LOGICAL;
 LOCAL
  x : SET OF product_definition_relationship;
  i : INTEGER;
  local_relatives : SET OF product_definition;
 END_LOCAL;
 REPEAT i := 1 TO HIINDEX(relatives);
  IF relation.relating_product_definition :=: relatives[i]
   THEN
    RETURN (FALSE);
  END_IF;
 END_REPEAT;
 x := bag_to_set(USEDIN(relation.relating_product_definition,specific_relation
      ));
 local_relatives := relatives + relation.relating_product_definition;
 IF SIZEOF(x) > 0
  THEN
   REPEAT i := 1 TO HIINDEX(x);
    IF NOT acyclic_product_definition_relationship(x[i],local_relatives,
       specific_relation)
     THEN
      RETURN (FALSE);
    END_IF;
   END_REPEAT;
 END_IF;
 RETURN (TRUE);
END_FUNCTION;


FUNCTION acyclic_solid_replica(rep : solid_replica;parent : solid_model) :
                                BOOLEAN;
 IF NOT (('AUTOMOTIVE_DESIGN.SOLID_REPLICA') IN TYPEOF(parent))
  THEN
   RETURN (TRUE);
 END_IF;
 IF (parent :=: rep)
  THEN
   RETURN (FALSE);
  ELSE
   RETURN (acyclic_solid_replica(rep,parent\solid_replica.parent_solid));
 END_IF;
END_FUNCTION;


FUNCTION acyclic_surface_replica(rep : surface_replica;parent : surface) :
                                 BOOLEAN;
 IF NOT (('AUTOMOTIVE_DESIGN.SURFACE_REPLICA') IN TYPEOF(parent))
  THEN
   RETURN (TRUE);
 END_IF;
 IF (parent :=: rep)
  THEN
   RETURN (FALSE);
```

```
  ELSE
   RETURN (acyclic_surface_replica(rep,parent\surface_replica.parent_surface))
    ;
 END_IF;
END_FUNCTION;


FUNCTION aspect_ratio(p : planar_box) : positive_ratio_measure;
 RETURN (p.size_in_x / p.size_in_y);
END_FUNCTION;


FUNCTION assembly_shape_is_defined(assy : next_assembly_usage_occurrence) :
                                  BOOLEAN;
 LOCAL
  sdr_set : SET OF shape_definition_representation := [];
  srr_set : SET OF shape_representation_relationship := [];
 END_LOCAL;
 sdr_set := QUERY(pdr <* bag_to_set(USEDIN(assy.related_product_definition,
            'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION_REPRESENTATION.' +
            'DEFINITION.DEFINITION')) |
            'AUTOMOTIVE_DESIGN.SHAPE_DEFINITION_REPRESENTATION' IN TYPEOF(pdr)
            ) + QUERY(pdr <* bag_to_set(USEDIN(assy.related_product_definition
            ,'AUTOMOTIVE_DESIGN.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION.' +
            'DEFINITION.RELATED_PRODUCT_DEFINITION')) |
            'AUTOMOTIVE_DESIGN.SHAPE_DEFINITION_REPRESENTATION' IN TYPEOF(pdr)
            );
 IF SIZEOF(sdr_set) > 0
  THEN
   REPEAT i := 1 TO HIINDEX(sdr_set);
    srr_set := QUERY(rr <* bag_to_set(USEDIN(sdr_set[i]\
               property_definition_representation.used_representation,
               'AUTOMOTIVE_DESIGN.REPRESENTATION_RELATIONSHIP.REP_2')) |
               'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_RELATIONSHIP' IN TYPEOF
               (rr));
    IF SIZEOF(srr_set) > 0
     THEN
      REPEAT j := 1 TO HIINDEX(srr_set);
       IF SIZEOF(QUERY(pdr <* bag_to_set(USEDIN(srr_set[j]\
          representation_relationship.rep_1,
          'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION_REPRESENTATION.' +
          'USED_REPRESENTATION')) |
          'AUTOMOTIVE_DESIGN.SHAPE_DEFINITION_REPRESENTATION' IN TYPEOF(pdr))
          * QUERY(pdr <* bag_to_set(USEDIN(assy.relating_product_definition,
          'AUTOMOTIVE_DESIGN.PROPERTY_DEFINITION_REPRESENTATION.' +
          'USED_REPRESENTATION.DEFINITION')) |
          'AUTOMOTIVE_DESIGN.SHAPE_DEFINITION_REPRESENTATION' IN TYPEOF(pdr)))
           >= 1
        THEN
         IF SIZEOF(QUERY(cdsr <* USEDIN(assy,'AUTOMOTIVE_DESIGN.' +
            'CONTEXT_DEPENDENT_SHAPE_REPRESENTATION.' +
            'REPRESENTED_PRODUCT_RELATION') | NOT ((cdsr.
```

```
            represented_product_relation.definition :=: assy) AND (cdsr.
            representation_relation :=: srr_set[j]))))) > 0
          THEN
           RETURN (FALSE);
         END_IF;
       END_IF;
      END_REPEAT;
    END_IF;
   END_REPEAT;
 END_IF;
 RETURN (TRUE);
END_FUNCTION;


FUNCTION associated_surface(arg : pcurve_or_surface) : surface;
 LOCAL
  surf : surface;
 END_LOCAL;
 IF 'AUTOMOTIVE_DESIGN.PCURVE' IN TYPEOF(arg)
  THEN
   surf := arg.basis_surface;
  ELSE
   surf := arg;
 END_IF;
 RETURN (surf);
END_FUNCTION;


FUNCTION bag_to_set(the_bag : BAG OF GENERIC : intype) : SET OF GENERIC :
                    intype;
 LOCAL
  the_set : SET OF GENERIC : intype := [];
  i : INTEGER;
 END_LOCAL;
 IF SIZEOF(the_bag) > 0
  THEN
   REPEAT i := 1 TO HIINDEX(the_bag);
    the_set := the_set + the_bag[i];
   END_REPEAT;
 END_IF;
 RETURN (the_set);
END_FUNCTION;


FUNCTION base_axis(dim : INTEGER;axis1,axis2,axis3 : direction) : LIST[2:3] OF
                   direction;
 LOCAL
  vec : direction;
  u : LIST[2:3] OF direction;
  factor : REAL;
 END_LOCAL;
 IF (dim = 3)
  THEN
```

```
    u[3] := NVL(normalise(axis3),direction([0.0,0.0,1.0]));
    u[1] := first_proj_axis(u[3],axis1);
    u[2] := second_proj_axis(u[3],u[1],axis2);
   ELSE
    u[3] := ?;
    IF EXISTS(axis1)
     THEN
      u[1] := normalise(axis1);
      u[2] := orthogonal_complement(u[1]);
      IF EXISTS(axis2)
       THEN
        factor := dot_product(axis2,u[2]);
        IF (factor < 0.0)
         THEN
          u[2].direction_ratios[1] := - u[2].direction_ratios[1];
          u[2].direction_ratios[2] := - u[2].direction_ratios[2];
        END_IF;
      END_IF;
     ELSE
      IF EXISTS(axis2)
       THEN
        u[2] := normalise(axis2);
        u[1] := orthogonal_complement(u[2]);
        u[1].direction_ratios[1] := - u[1].direction_ratios[1];
        u[1].direction_ratios[2] := - u[1].direction_ratios[2];
       ELSE
        u[1].direction_ratios[1] := 1.0;
        u[1].direction_ratios[2] := 0.0;
        u[2].direction_ratios[1] := 0.0;
        u[2].direction_ratios[2] := 1.0;
      END_IF;
    END_IF;
  END_IF;
  RETURN (u);
END_FUNCTION;


FUNCTION basis_curve_check(cv : curve) : BOOLEAN;
  IF (('AUTOMOTIVE_DESIGN.B_SPLINE_CURVE' IN TYPEOF(cv)) AND (cv\b_spline_curve
     .self_intersect = FALSE))
   THEN
    RETURN (TRUE);
   ELSE
    IF SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE','AUTOMOTIVE_DESIGN.CONIC',
       'AUTOMOTIVE_DESIGN.LINE'] * TYPEOF(cv)) = 1
     THEN
      RETURN (TRUE);
     ELSE
      IF SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE',
         'AUTOMOTIVE_DESIGN.CURVE_REPLICA'] * TYPEOF(cv)) = 1
       THEN
        RETURN (basis_curve_check(cv\curve_replica.parent_curve));
       ELSE
```

```
IF ((SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE',
   'AUTOMOTIVE_DESIGN.OFFSET_CURVE_3D'] * TYPEOF(cv)) = 1) AND (cv\
   offset_curve_3d.self_intersect = FALSE))
 THEN
  RETURN (basis_curve_check(cv\offset_curve_3d.basis_curve));
 ELSE
  IF SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE',
     'AUTOMOTIVE_DESIGN.PCURVE'] * TYPEOF(cv)) = 1
   THEN
    RETURN ((basis_curve_check(cv\pcurve.reference_to_curve\
     representation.items[1])) AND (basis_surface_check(cv\pcurve.
     basis_surface)));
   ELSE
    IF SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE',
       'AUTOMOTIVE_DESIGN.SURFACE_CURVE'] * TYPEOF(cv)) = 1
     THEN
      IF basis_curve_check(cv\surface_curve.curve_3d)
       THEN
        REPEAT i := 1 TO SIZEOF(cv\surface_curve.associated_geometry);
         IF 'AUTOMOTIVE_DESIGN.SURFACE' IN TYPEOF(cv\surface_curve.
            associated_geometry[i])
          THEN
           IF NOT basis_surface_check(cv\surface_curve.
              associated_geometry[i])
            THEN
             RETURN (FALSE);
           END_IF;
          ELSE
           IF 'AUTOMOTIVE_DESIGN.PCURVE' IN TYPEOF(cv\surface_curve.
              associated_geometry[i])
            THEN
             IF NOT basis_curve_check(cv\surface_curve.
                associated_geometry[i])
              THEN
               RETURN (FALSE);
             END_IF;
           END_IF;
         END_IF;
        END_REPEAT;
        RETURN (TRUE);
      END_IF;
     ELSE
      IF 'AUTOMOTIVE_DESIGN.POLYLINE' IN TYPEOF(cv)
       THEN
        IF (SIZEOF(cv\polyline.points) >= 3)
         THEN
          RETURN (TRUE);
        END_IF;
      END_IF;
    END_IF;
  END_IF;
END_IF;
```

```
     END_IF;
   END_IF;
 END_IF;
 RETURN (FALSE);
END_FUNCTION;


FUNCTION basis_surface_check(surf : surface) : BOOLEAN;
 IF 'AUTOMOTIVE_DESIGN.ELEMENTARY_SURFACE' IN TYPEOF(surf)
  THEN
   RETURN (TRUE);
  ELSE
   IF 'AUTOMOTIVE_DESIGN.SWEPT_SURFACE' IN TYPEOF(surf)
    THEN
     RETURN (basis_curve_check(surf\swept_surface.swept_curve));
    ELSE
     IF (('AUTOMOTIVE_DESIGN.OFFSET_SURFACE' IN TYPEOF(surf)) AND (surf\
        offset_surface.self_intersect = FALSE))
      THEN
       RETURN (basis_surface_check(surf\offset_surface.basis_surface));
      ELSE
       IF 'AUTOMOTIVE_DESIGN.SURFACE_REPLICA' IN TYPEOF(surf)
        THEN
         RETURN (basis_surface_check(surf\surface_replica.parent_surface));
        ELSE
         IF (('AUTOMOTIVE_DESIGN.B_SPLINE_SURFACE' IN TYPEOF(surf)) AND (surf\
            b_spline_surface.self_intersect = FALSE))
          THEN
           RETURN (TRUE);
         END_IF;
       END_IF;
     END_IF;
   END_IF;
 END_IF;
 RETURN (FALSE);
END_FUNCTION;


FUNCTION boolean_choose(b : BOOLEAN;choice1,choice2 : GENERIC : item) : GENERIC
                        : item;
 IF b
  THEN
   RETURN (choice1);
  ELSE
   RETURN (choice2);
 END_IF;
END_FUNCTION;


FUNCTION build_2axes(ref_direction : direction) : LIST[2:2] OF direction;
 LOCAL
  u : LIST[2:2] OF direction;
 END_LOCAL;
```

```
 u[1] := NVL(normalise(ref_direction),direction([1.0,0.0]));
 u[2] := orthogonal_complement(u[1]);
 RETURN (u);
END_FUNCTION;


FUNCTION build_axes(axis,ref_direction : direction) : LIST[3:3] OF direction;
 LOCAL
  u : LIST[3:3] OF direction;
 END_LOCAL;
 u[3] := NVL(normalise(axis),direction([0.0,0.0,1.0]));
 u[1] := first_proj_axis(u[3],ref_direction);
 u[2] := normalise(cross_product(u[3],u[1])).orientation;
 RETURN (u);
END_FUNCTION;


FUNCTION check_associative_shape_aspects(sdr : shape_definition_representation
                                          ) : BOOLEAN;
 LOCAL
  sr1,sr2 : SET OF representation;
  dm,pv : SET OF representation;
  i,j,k : INTEGER;
 END_LOCAL;
 IF (SIZEOF(sdr.used_representation.items) <> 2)
  THEN
   RETURN (FALSE);
 END_IF;
 sr1 := using_representations(sdr.used_representation.items[1]);
 sr2 := using_representations(sdr.used_representation.items[2]);
 REPEAT i := 1 TO HIINDEX(sr1);
  dm := representations_mapped_into(sr1[i]);
  REPEAT j := 1 TO HIINDEX(dm);
   IF ('AUTOMOTIVE_DESIGN.DRAUGHTING_MODEL' IN TYPEOF(dm[j]))
    THEN
     IF (dm[j] IN sr2)
      THEN
       RETURN (TRUE);
     END_IF;
     pv := representations_mapped_into(dm[j]);
     REPEAT k := 1 TO HIINDEX(pv);
      IF ('AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN TYPEOF(pv[k]))
       THEN
        IF (pv[k] IN sr2)
         THEN
          RETURN (TRUE);
        END_IF;
      END_IF;
     END_REPEAT;
   END_IF;
  END_REPEAT;
 END_REPEAT;
 REPEAT i := 1 TO HIINDEX(sr2);
```

```
  dm := representations_mapped_into(sr2[i]);
  REPEAT j := 1 TO HIINDEX(dm);
   IF ('AUTOMOTIVE_DESIGN.DRAUGHTING_MODEL' IN TYPEOF(dm[j]))
    THEN
     IF (dm[j] IN sr1)
      THEN
       RETURN (TRUE);
     END_IF;
     pv := representations_mapped_into(dm[j]);
     REPEAT k := 1 TO HIINDEX(pv);
      IF ('AUTOMOTIVE_DESIGN.PRESENTATION_VIEW' IN TYPEOF(pv[k]))
       THEN
        IF (pv[k] IN sr1)
         THEN
          RETURN (TRUE);
        END_IF;
      END_IF;
     END_REPEAT;
   END_IF;
  END_REPEAT;
 END_REPEAT;
 RETURN (FALSE);
END_FUNCTION;


FUNCTION check_text_alignment(ct : composite_text) : BOOLEAN;
 LOCAL
  a : SET OF text_alignment := [];
  i : INTEGER;
 END_LOCAL;
 REPEAT i := 1 TO HIINDEX(ct.collected_text);
  a := a + [ct.collected_text[i]\text_literal.alignment];
 END_REPEAT;
 RETURN (SIZEOF(a) = 1);
END_FUNCTION;


FUNCTION check_text_font(ct : composite_text) : BOOLEAN;
 LOCAL
  f : SET OF font_select := [];
  i : INTEGER;
 END_LOCAL;
 REPEAT i := 1 TO HIINDEX(ct.collected_text);
  f := f + [ct.collected_text[i]\text_literal.font];
 END_REPEAT;
 RETURN (SIZEOF(f) <= 1);
END_FUNCTION;


FUNCTION conditional_reverse(p : BOOLEAN;an_item : reversible_topology) :
                             reversible_topology;
 IF p
  THEN
```

```
    RETURN (an_item);
   ELSE
    RETURN (topology_reversed(an_item));
  END_IF;
END_FUNCTION;


FUNCTION constraints_composite_curve_on_surface(c : composite_curve_on_surface
                                                ) : BOOLEAN;
 LOCAL
  n_segments : INTEGER := SIZEOF(c.segments);
 END_LOCAL;
 REPEAT k := 1 TO n_segments;
  IF (NOT ('AUTOMOTIVE_DESIGN.PCURVE' IN TYPEOF(c\composite_curve.segments[k].
     parent_curve))) AND (NOT ('AUTOMOTIVE_DESIGN.SURFACE_CURVE' IN TYPEOF(c\
     composite_curve.segments[k].parent_curve))) AND (NOT (
     'AUTOMOTIVE_DESIGN.COMPOSITE_CURVE_ON_SURFACE' IN TYPEOF(c\
     composite_curve.segments[k].parent_curve)))
   THEN
    RETURN (FALSE);
  END_IF;
 END_REPEAT;
 RETURN (TRUE);
END_FUNCTION;


FUNCTION constraints_geometry_shell_based_surface_model(m :
         shell_based_surface_model) : BOOLEAN;
 LOCAL
  result : BOOLEAN := TRUE;
 END_LOCAL;
 REPEAT j := 1 TO SIZEOF(m.sbsm_boundary);
  IF (NOT ('AUTOMOTIVE_DESIGN.OPEN_SHELL' IN TYPEOF(m.sbsm_boundary[j])) AND (
     NOT ('AUTOMOTIVE_DESIGN.CLOSED_SHELL' IN TYPEOF(m.sbsm_boundary[j]))))
   THEN
    result := FALSE;
    RETURN (result);
  END_IF;
 END_REPEAT;
 RETURN (result);
END_FUNCTION;


FUNCTION constraints_param_b_spline(degree,up_knots,up_cp : INTEGER;knot_mult
                                    : LIST OF INTEGER;knots : LIST OF
                                    parameter_value) : BOOLEAN;
 LOCAL
  result : BOOLEAN := TRUE;
  k,l,sum : INTEGER;
 END_LOCAL;
 sum := knot_mult[1];
 REPEAT i := 2 TO up_knots;
  sum := sum + knot_mult[i];
```

```
 END_REPEAT;
 IF (degree < 1) OR (up_knots < 2) OR (up_cp < degree) OR (sum <> (degree +
    up_cp + 2))
  THEN
   result := FALSE;
   RETURN (result);
 END_IF;
 k := knot_mult[1];
 IF (k < 1) OR (k > degree + 1)
  THEN
   result := FALSE;
   RETURN (result);
 END_IF;
 REPEAT i := 2 TO up_knots;
  IF (knot_mult[i] < 1) OR (knots[i] <= knots[i - 1])
   THEN
    result := FALSE;
    RETURN (result);
  END_IF;
  k := knot_mult[i];
  IF (i < up_knots) AND (k > degree)
   THEN
    result := FALSE;
    RETURN (result);
  END_IF;
  IF (i = up_knots) AND (k > degree + 1)
   THEN
    result := FALSE;
    RETURN (result);
  END_IF;
 END_REPEAT;
 RETURN (result);
END_FUNCTION;


FUNCTION constraints_rectangular_composite_surface(s :
         rectangular_composite_surface) : BOOLEAN;
 REPEAT i := 1 TO s.n_u;
  REPEAT j := 1 TO s.n_v;
   IF NOT (('AUTOMOTIVE_DESIGN.B_SPLINE_SURFACE' IN TYPEOF(s.segments[i][j].
      parent_surface)) OR ('AUTOMOTIVE_DESIGN.RECTANGULAR_TRIMMED_SURFACE' IN
      TYPEOF(s.segments[i][j].parent_surface)))
    THEN
     RETURN (FALSE);
   END_IF;
  END_REPEAT;
 END_REPEAT;
 REPEAT i := 1 TO s.n_u - 1;
  REPEAT j := 1 TO s.n_v;
   IF s.segments[i][j].u_transition = discontinuous
    THEN
     RETURN (FALSE);
   END_IF;
```

```
   END_REPEAT;
  END_REPEAT;
  REPEAT i := 1 TO s.n_u;
   REPEAT j := 1 TO s.n_v - 1;
    IF s.segments[i][j].v_transition = discontinuous
     THEN
      RETURN (FALSE);
    END_IF;
   END_REPEAT;
  END_REPEAT;
  RETURN (TRUE);
 END_FUNCTION;


 FUNCTION convert_plane_angle_for_pair_from_radian(pair : kinematic_pair;
                                                   angle_expr : REAL) :
                                                   plane_angle_measure;
  LOCAL
   link_rep : kinematic_link_representation := representation_of_link(pair.
   joint.first_link);
   link_cntxt : representation_context;
   pa_units : SET OF unit := [];
   pau : unit;
   conv_factor : REAL := 1.0;
   result : plane_angle_measure;
  END_LOCAL;
  link_cntxt := link_rep\representation.context_of_items;
  IF NOT ('AUTOMOTIVE_DESIGN.GLOBAL_UNIT_ASSIGNED_CONTEXT' IN TYPEOF(link_cntxt
     ))
   THEN
    RETURN (?);
  END_IF;
  pa_units := QUERY(unit <* link_cntxt\global_unit_assigned_context.units |
             'AUTOMOTIVE_DESIGN.PLANE_ANGLE_UNIT' IN TYPEOF(unit));
  IF SIZEOF(pa_units) <> 1
   THEN
    RETURN (?);
  END_IF;
  pau := pa_units[1];
  IF (NOT ('AUTOMOTIVE_DESIGN.SI_UNIT' IN TYPEOF(pau)) AND NOT (
     'AUTOMOTIVE_DESIGN.CONVERSION_BASED_UNIT' IN TYPEOF(pau)))
   THEN
    RETURN (?);
  END_IF;
  REPEAT WHILE ('AUTOMOTIVE_DESIGN.CONVERSION_BASED_UNIT' IN TYPEOF(pau));
   conv_factor := conv_factor * pau\conversion_based_unit.conversion_factor.
                  value_component;
   pau := pau\conversion_based_unit.conversion_factor.unit_component;
   IF ((NOT ('AUTOMOTIVE_DESIGN.SI_UNIT' IN TYPEOF(pau)) AND NOT (
      'AUTOMOTIVE_DESIGN.CONVERSION_BASED_UNIT' IN TYPEOF(pau))) OR (NOT (
      'AUTOMOTIVE_DESIGN.PLANE_ANGLE_UNIT' IN TYPEOF(pau))))
    THEN
     RETURN (?);
```

```
  END_IF;
 END_REPEAT;
 IF (pau\si_unit.name <> si_unit_name.radian)
  THEN
   RETURN (?);
 END_IF;
 CASE pau\si_unit.prefix OF
  si_prefix.exa : conv_factor := 1.E18 * conv_factor;
  si_prefix.peta : conv_factor := 1.E15 * conv_factor;
  si_prefix.tera : conv_factor := 1.E12 * conv_factor;
  si_prefix.giga : conv_factor := 1.E9 * conv_factor;
  si_prefix.mega : conv_factor := 1.E6 * conv_factor;
  si_prefix.kilo : conv_factor := 1.E3 * conv_factor;
  si_prefix.hecto : conv_factor := 1.E2 * conv_factor;
  si_prefix.deca : conv_factor := 1.E1 * conv_factor;
  si_prefix.deci : conv_factor := 1.E-1 * conv_factor;
  si_prefix.centi : conv_factor := 1.E-2 * conv_factor;
  si_prefix.milli : conv_factor := 1.E-3 * conv_factor;
  si_prefix.micro : conv_factor := 1.E-6 * conv_factor;
  si_prefix.nano : conv_factor := 1.E-9 * conv_factor;
  si_prefix.pico : conv_factor := 1.E-12 * conv_factor;
  si_prefix.femto : conv_factor := 1.E-15 * conv_factor;
  si_prefix.atto : conv_factor := 1.E-18 * conv_factor;
 END_CASE;
 result := angle_expr / conv_factor;
 RETURN (result);
END_FUNCTION;


FUNCTION convert_spatial_to_ypr_rotation(pair : kinematic_pair;rotation :
                                          spatial_rotation) : ypr_rotation;
 LOCAL
  axis : direction;
  angle : plane_angle_measure;
  conv_angle : plane_angle_measure;
  ya,pa,ra : plane_angle_measure;
  ucf : REAL;
  dx,dy,dz : REAL;
  s_a,c_a : REAL;
  rotmat : ARRAY[1:3] OF ARRAY[1:3] OF REAL;
  cm1 : REAL;
  s_y,c_y : REAL;
  s_r,c_r : REAL;
 END_LOCAL;
 IF 'AUTOMOTIVE_DESIGN.YPR_ROTATION' IN TYPEOF(rotation)
  THEN
   RETURN (rotation);
 END_IF;
 axis := normalise(rotation\rotation_about_direction.direction_of_axis);
 angle := rotation\rotation_about_direction.rotation_angle;
 IF (angle = 0.0)
  THEN
   RETURN ([0.0,0.0,0.0]);
```

```
END_IF;
dx := axis.direction_ratios[1];
dy := axis.direction_ratios[2];
dz := axis.direction_ratios[3];
conv_angle := plane_angle_for_pair_in_radian(pair,angle);
IF (conv_angle = ?)
 THEN
  RETURN (?);
END_IF;
ucf := angle / conv_angle;
s_a := SIN(conv_angle);
c_a := COS(conv_angle);
IF (dy = 0.0) AND (dx * dz = 0.0)
 THEN
  REPEAT WHILE (conv_angle <= - PI);
   conv_angle := conv_angle + 2.0 * PI;
  END_REPEAT;
  REPEAT WHILE (conv_angle > PI);
   conv_angle := conv_angle - 2.0 * PI;
  END_REPEAT;
  ya := ucf * conv_angle;
  IF (conv_angle <> PI)
   THEN
    ra := - ya;
   ELSE
    ra := ya;
  END_IF;
  IF (dx <> 0.0)
   THEN
    IF (dx > 0.0)
     THEN
      RETURN ([0.0,0.0,ya]);
     ELSE
      RETURN ([0.0,0.0,ra]);
    END_IF;
   ELSE
    IF (dz > 0.0)
     THEN
      RETURN ([ya,0.0,0.0]);
     ELSE
      RETURN ([ra,0.0,0.0]);
    END_IF;
  END_IF;
END_IF;
IF ((dy <> 0.0) AND (dx = 0.0) AND (dz = 0.0))
 THEN
  IF (c_a >= 0.0)
   THEN
    ya := 0.0;
    ra := 0.0;
   ELSE
    ya := ucf * PI;
```

```
    ra := ya;
  END_IF;
  pa := ucf * ATAN(s_a,ABS(c_a));
  IF (dy < 0.0)
   THEN
    pa := - pa;
  END_IF;
  RETURN ([ya,pa,ra]);
END_IF;
cm1 := 1.0 - c_a;
rotmat := [[dx * dx * cm1 + c_a,dx * dy * cm1 - dz * s_a,dx * dz * cm1 + dy *
           s_a],[dx * dy * cm1 + dz * s_a,dy * dy * cm1 + c_a,dy * dz * cm1 -
           dx * s_a],[dx * dz * cm1 - dy * s_a,dy * dz * cm1 + dx * s_a,dz *
          dz * cm1 + c_a]];
IF (ABS(rotmat[1][3]) = 1.0)
 THEN
  BEGIN
   IF (rotmat[1][3] = 1.0)
    THEN
     pa := 0.5 * PI;
    ELSE
     pa := - 0.5 * PI;
   END_IF;
   ra := 0.0;
   ya := ATAN(rotmat[2][1],rotmat[2][2]);
   IF (rotmat[2][2] < 0.0)
    THEN
     IF ya <= 0.0
      THEN
       ya := ya + PI;
      ELSE
       ya := ya - PI;
     END_IF;
   END_IF;
  END;
 ELSE
  BEGIN
   ya := ATAN(- rotmat[1][2],rotmat[1][1]);
   IF (rotmat[1][1] < 0.0)
    THEN
     IF (ya <= 0.0)
      THEN
       ya := ya + PI;
      ELSE
       ya := ya - PI;
     END_IF;
   END_IF;
   ra := ATAN(- rotmat[2][3],rotmat[3][3]);
   IF (rotmat[3][3] < 0.0)
    THEN
     IF (ra <= 0.0)
      THEN
```

```
        ra := ra + PI;
       ELSE
        ra := ra - PI;
      END_IF;
    END_IF;
    s_y := SIN(ya);
    c_y := COS(ya);
    s_r := SIN(ra);
    c_r := COS(ra);
    IF ((ABS(s_y) > ABS(c_y)) AND (ABS(s_y) > ABS(s_r)) AND (ABS(s_y) > ABS(
       c_r)))
     THEN
      cm1 := - rotmat[1][2] / s_y;
     ELSE
      IF ((ABS(c_y) > ABS(s_r)) AND (ABS(c_y) > ABS(c_r)))
       THEN
        cm1 := rotmat[1][1] / c_y;
       ELSE
        IF (ABS(s_r) > ABS(c_r))
         THEN
          cm1 := - rotmat[2][3] / s_r;
         ELSE
          cm1 := rotmat[3][3] / c_r;
        END_IF;
      END_IF;
    END_IF;
    pa := ATAN(rotmat[1][3],cm1);
   END;
 END_IF;
 ya := ya * ucf;
 pa := pa * ucf;
 ra := ra * ucf;
 RETURN ([ya,pa,ra]);
END_FUNCTION;


FUNCTION coordinated_pair_link_representation(link : kinematic_link;
                                              pair_placement : rigid_placement
                                              ) : BOOLEAN;
 LOCAL
  link_rep : kinematic_link_representation;
 END_LOCAL;
 link_rep := representation_of_link(link);
 IF (link_rep = ?)
  THEN
   RETURN (FALSE);
  ELSE
   IF NOT (pair_placement IN link_rep\representation.items)
    THEN
     RETURN (FALSE);
    ELSE
     RETURN (TRUE);
   END_IF;
```

```
 END_IF;
END_FUNCTION;


FUNCTION cross_product(arg1,arg2 : direction) : vector;
 LOCAL
  mag : REAL;
  res : direction;
  v1,v2 : LIST[3:3] OF REAL;
  result : vector;
 END_LOCAL;
 IF (NOT EXISTS(arg1) OR (arg1.dim = 2)) OR (NOT EXISTS(arg2) OR (arg2.dim = 2
    ))
  THEN
   RETURN (?);
  ELSE
   BEGIN
    v1 := normalise(arg1).direction_ratios;
    v2 := normalise(arg2).direction_ratios;
    res.direction_ratios[1] := (v1[2] * v2[3] - v1[3] * v2[2]);
    res.direction_ratios[2] := (v1[3] * v2[1] - v1[1] * v2[3]);
    res.direction_ratios[3] := (v1[1] * v2[2] - v1[2] * v2[1]);
    mag := 0.0;
    REPEAT i := 1 TO 3;
     mag := mag + res.direction_ratios[i] * res.direction_ratios[i];
    END_REPEAT;
    IF (mag > 0.0)
     THEN
      result.orientation := res;
      result.magnitude := SQRT(mag);
     ELSE
      result.orientation := arg1;
      result.magnitude := 0.0;
    END_IF;
    RETURN (result);
   END;
 END_IF;
END_FUNCTION;


FUNCTION curve_weights_positive(b : rational_b_spline_curve) : BOOLEAN;
 LOCAL
  result : BOOLEAN := TRUE;
 END_LOCAL;
 REPEAT i := 0 TO b.upper_index_on_control_points;
  IF b.weights[i] <= 0.0
   THEN
    result := FALSE;
    RETURN (result);
  END_IF;
 END_REPEAT;
 RETURN (result);
END_FUNCTION;
```

```
FUNCTION derive_dimensional_exponents(x : unit) : dimensional_exponents;
 LOCAL
  i : INTEGER;
  result : dimensional_exponents := dimensional_exponents(0.0,0.0,0.0,0.0,0.0,
  0.0,0.0);
 END_LOCAL;
 IF 'AUTOMOTIVE_DESIGN.DERIVED_UNIT' IN TYPEOF(x)
  THEN
   REPEAT i := LOINDEX(x.elements) TO HIINDEX(x.elements);
    result.length_exponent := result.length_exponent + (x.elements[i].exponent
                              * x.elements[i].unit.dimensions.length_exponent
                             );
    result.mass_exponent := result.mass_exponent + (x.elements[i].exponent * x
                            .elements[i].unit.dimensions.mass_exponent);
    result.time_exponent := result.time_exponent + (x.elements[i].exponent * x
                            .elements[i].unit.dimensions.time_exponent);
    result.electric_current_exponent := result.electric_current_exponent + (x.
                                        elements[i].exponent * x.elements[i].
                                        unit.dimensions.
                                        electric_current_exponent);
    result.thermodynamic_temperature_exponent := result.
    thermodynamic_temperature_exponent + (x.elements[i].exponent * x.elements[
                                                 i].unit.dimensions.
    thermodynamic_temperature_exponent);
    result.amount_of_substance_exponent := result.amount_of_substance_exponent
                                            + (x.elements[i].exponent * x.
                                           elements[i].unit.dimensions.
                                           amount_of_substance_exponent);
    result.luminous_intensity_exponent := result.luminous_intensity_exponent +
                                           (x.elements[i].exponent * x.
                                          elements[i].unit.dimensions.
                                          luminous_intensity_exponent);
   END_REPEAT;
  ELSE
   result := x.dimensions;
 END_IF;
 RETURN (result);
END_FUNCTION;


FUNCTION dimension_of(item : geometric_representation_item) : dimension_count;
 LOCAL
  x : SET OF representation;
  y : representation_context;
 END_LOCAL;
 x := using_representations(item);
 y := x[1].context_of_items;
 RETURN (y\geometric_representation_context.coordinate_space_dimension);
END_FUNCTION;


FUNCTION dimensions_for_si_unit(n : si_unit_name) : dimensional_exponents;
```

```
 CASE n OF
  metre : RETURN (dimensional_exponents(1.0,0.0,0.0,0.0,0.0,0.0,0.0));
  gram : RETURN (dimensional_exponents(0.0,1.0,0.0,0.0,0.0,0.0,0.0));
  second : RETURN (dimensional_exponents(0.0,0.0,1.0,0.0,0.0,0.0,0.0));
  ampere : RETURN (dimensional_exponents(0.0,0.0,0.0,1.0,0.0,0.0,0.0));
  kelvin : RETURN (dimensional_exponents(0.0,0.0,0.0,0.0,1.0,0.0,0.0));
  mole : RETURN (dimensional_exponents(0.0,0.0,0.0,0.0,0.0,1.0,0.0));
  candela : RETURN (dimensional_exponents(0.0,0.0,0.0,0.0,0.0,0.0,1.0));
  radian : RETURN (dimensional_exponents(0.0,0.0,0.0,0.0,0.0,0.0,0.0));
  steradian : RETURN (dimensional_exponents(0.0,0.0,0.0,0.0,0.0,0.0,0.0));
  hertz : RETURN (dimensional_exponents(0.0,0.0,- 1.0,0.0,0.0,0.0,0.0));
  newton : RETURN (dimensional_exponents(1.0,1.0,- 2.0,0.0,0.0,0.0,0.0));
  pascal : RETURN (dimensional_exponents(- 1.0,1.0,- 2.0,0.0,0.0,0.0,0.0));
  joule : RETURN (dimensional_exponents(2.0,1.0,- 2.0,0.0,0.0,0.0,0.0));
  watt : RETURN (dimensional_exponents(2.0,1.0,- 3.0,0.0,0.0,0.0,0.0));
  coulomb : RETURN (dimensional_exponents(0.0,0.0,1.0,1.0,0.0,0.0,0.0));
  volt : RETURN (dimensional_exponents(2.0,1.0,- 3.0,- 1.0,0.0,0.0,0.0));
  farad : RETURN (dimensional_exponents(- 2.0,- 1.0,4.0,1.0,0.0,0.0,0.0));
  ohm : RETURN (dimensional_exponents(2.0,1.0,- 3.0,- 2.0,0.0,0.0,0.0));
  siemens : RETURN (dimensional_exponents(- 2.0,- 1.0,3.0,2.0,0.0,0.0,0.0));
  weber : RETURN (dimensional_exponents(2.0,1.0,- 2.0,- 1.0,0.0,0.0,0.0));
  tesla : RETURN (dimensional_exponents(0.0,1.0,- 2.0,- 1.0,0.0,0.0,0.0));
  henry : RETURN (dimensional_exponents(2.0,1.0,- 2.0,- 2.0,0.0,0.0,0.0));
  degree_celsius : RETURN (dimensional_exponents(0.0,0.0,0.0,0.0,1.0,0.0,0.0))
                    ;
  lumen : RETURN (dimensional_exponents(0.0,0.0,0.0,0.0,0.0,0.0,1.0));
  lux : RETURN (dimensional_exponents(- 2.0,0.0,0.0,0.0,0.0,0.0,1.0));
  becquerel : RETURN (dimensional_exponents(0.0,0.0,- 1.0,0.0,0.0,0.0,0.0));
  gray : RETURN (dimensional_exponents(2.0,0.0,- 2.0,0.0,0.0,0.0,0.0));
  sievert : RETURN (dimensional_exponents(2.0,0.0,- 2.0,0.0,0.0,0.0,0.0));
 END_CASE;
END_FUNCTION;


FUNCTION dot_product(arg1,arg2 : direction) : REAL;
 LOCAL
  scalar : REAL;
  vec1,vec2 : direction;
  ndim : INTEGER;
 END_LOCAL;
 IF NOT EXISTS(arg1) OR NOT EXISTS(arg2)
  THEN
   scalar := ?;
  ELSE
   IF (arg1.dim <> arg2.dim)
    THEN
     scalar := ?;
    ELSE
     BEGIN
      vec1 := normalise(arg1);
      vec2 := normalise(arg2);
      ndim := arg1.dim;
      scalar := 0.0;
```

```
      REPEAT i := 1 TO ndim;
       scalar := scalar + vec1.direction_ratios[i] * vec2.direction_ratios[i];
      END_REPEAT;
     END;
   END_IF;
 END_IF;
 RETURN (scalar);
END_FUNCTION;


FUNCTION edge_reversed(an_edge : edge) : edge;
 LOCAL
  the_reverse : edge;
 END_LOCAL;
 IF ('AUTOMOTIVE_DESIGN.ORIENTED_EDGE' IN TYPEOF(an_edge))
  THEN
   the_reverse := oriented_edge(an_edge\oriented_edge.edge_element,(NOT (
                  an_edge\oriented_edge.orientation)));
  ELSE
   the_reverse := oriented_edge(an_edge,FALSE);
 END_IF;
 RETURN (the_reverse);
END_FUNCTION;


FUNCTION face_bound_reversed(a_face_bound : face_bound) : face_bound;
 LOCAL
  the_reverse : face_bound;
 END_LOCAL;
 IF ('AUTOMOTIVE_DESIGN.FACE_OUTER_BOUND' IN TYPEOF(a_face_bound))
  THEN
   the_reverse := face_bound(a_face_bound\face_bound.bound,(NOT (a_face_bound\
                  face_bound.orientation)));
  ELSE
   the_reverse := face_bound(a_face_bound.bound,(NOT (a_face_bound.orientation
                  )));
 END_IF;
 RETURN (the_reverse);
END_FUNCTION;


FUNCTION face_reversed(a_face : face) : face;
 LOCAL
  the_reverse : face;
 END_LOCAL;
 IF ('AUTOMOTIVE_DESIGN.ORIENTED_FACE' IN TYPEOF(a_face))
  THEN
   the_reverse := oriented_face(a_face\oriented_face.face_element,(NOT (a_face
                  \oriented_face.orientation)));
  ELSE
   the_reverse := oriented_face(a_face,FALSE);
 END_IF;
 RETURN (the_reverse);
```

```
END_FUNCTION;


FUNCTION first_proj_axis(z_axis,arg : direction) : direction;
 LOCAL
  x_axis : direction;
  v : direction;
  z : direction;
  x_vec : vector;
 END_LOCAL;
 IF (NOT EXISTS(z_axis)) OR (NOT EXISTS(arg)) OR (arg.dim <> 3)
  THEN
   x_axis := ?;
  ELSE
   z_axis := normalise(z_axis);
   IF NOT EXISTS(arg)
    THEN
     IF (z_axis <> direction([1.0,0.0,0.0]))
      THEN
       v := direction([1.0,0.0,0.0]);
      ELSE
       v := direction([0.0,1.0,0.0]);
     END_IF;
    ELSE
     IF ((cross_product(arg,z).magnitude) = 0.0)
      THEN
       RETURN (?);
      ELSE
       v := normalise(arg);
     END_IF;
   END_IF;
   x_vec := scalar_times_vector(dot_product(v,z),z_axis);
   x_axis := vector_difference(v,x_vec).orientation;
   x_axis := normalise(x_axis);
 END_IF;
 RETURN (x_axis);
END_FUNCTION;


FUNCTION frame_associated_to_background(frame : rigid_placement;background :
                                        kinematic_frame_background) : BOOLEAN;
 LOCAL
  rep_bag : BAG OF kinematic_frame_background_representation;
  trf_bag : BAG OF kinematic_frame_based_transformation;
  trm_bag : BAG OF kinematic_frame_based_transformation;
  ass_bag : BAG OF kinematic_frame_background_representation_association;
  rep : kinematic_frame_background_representation;
  ass : kinematic_frame_background_representation_association;
 END_LOCAL;
 rep_bag := USEDIN(background\representation_item,'AUTOMOTIVE_DESIGN.' +
            'KINEMATIC_FRAME_BACKGROUND_REPRESENTATION\' +
            'REPRESENTATION.ITEMS');
 IF SIZEOF(rep_bag) = 0
```

```
   THEN
    RETURN (FALSE);
  END_IF;
  trf_bag := USEDIN(frame,'AUTOMOTIVE_DESIGN.' +
             'KINEMATIC_FRAME_BASED_TRANSFORMATION.' + 'TRANSFORMER_FRAME');
  IF SIZEOF(trf_bag) = 0
   THEN
    RETURN (FALSE);
  END_IF;
  REPEAT i := 1 TO HIINDEX(rep_bag);
   rep := rep_bag[i];
   ass_bag := USEDIN(rep\representation,'AUTOMOTIVE_DESIGN.' +
              'KINEMATIC_FRAME_BACKGROUND_REPRESENTATION_ASSOCIATION\' +
              'REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION\' +
              'REPRESENTATION_RELATIONSHIP.REP_2');
   IF SIZEOF(ass_bag) > 0
    THEN
     REPEAT j := 1 TO HIINDEX(ass_bag);
      ass := ass_bag[j];
      trm_bag := QUERY(trm <* trf_bag | (trm\
                 functionally_defined_transformation :=: ass\
                 representation_relationship_with_transformation.
                 transformation_operator));
      IF SIZEOF(trm_bag) > 0
       THEN
        RETURN (TRUE);
      END_IF;
     END_REPEAT;
   END_IF;
  END_REPEAT;
  RETURN (FALSE);
 END_FUNCTION;


 FUNCTION gbsf_check_curve(cv : curve) : BOOLEAN;
  IF SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE','AUTOMOTIVE_DESIGN.CIRCLE',
     'AUTOMOTIVE_DESIGN.ELLIPSE'] * TYPEOF(cv)) = 1
   THEN
    RETURN (TRUE);
   ELSE
    IF (('AUTOMOTIVE_DESIGN.B_SPLINE_CURVE' IN TYPEOF(cv)) AND (cv\
       b_spline_curve.self_intersect = FALSE))
     THEN
      RETURN (TRUE);
     ELSE
      IF (('AUTOMOTIVE_DESIGN.COMPOSITE_CURVE' IN TYPEOF(cv)) AND (cv\
         composite_curve.self_intersect = FALSE))
       THEN
        RETURN (SIZEOF(QUERY(seg <* cv\composite_curve.segments | NOT (
         gbsf_check_curve(seg.parent_curve)))) = 0);
       ELSE
        IF SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE',
           'AUTOMOTIVE_DESIGN.CURVE_REPLICA'] * TYPEOF(cv)) = 1
```

```
THEN
 RETURN (gbsf_check_curve(cv\curve_replica.parent_curve));
ELSE
 IF ((SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE',
    'AUTOMOTIVE_DESIGN.OFFSET_CURVE_3D'] * TYPEOF(cv)) = 1) AND (cv\
    offset_curve_3d.self_intersect = FALSE))
  THEN
   RETURN (gbsf_check_curve(cv\offset_curve_3d.basis_curve));
  ELSE
   IF SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE',
      'AUTOMOTIVE_DESIGN.PCURVE'] * TYPEOF(cv)) = 1
    THEN
     RETURN ((gbsf_check_curve(cv\pcurve.reference_to_curve.items[1]))
       AND (gbsf_check_surface(cv\pcurve.basis_surface)));
    ELSE
     IF 'AUTOMOTIVE_DESIGN.POLYLINE' IN TYPEOF(cv)
      THEN
       IF (SIZEOF(cv\polyline.points) >= 3) AND (SIZEOF((bag_to_set(
          USEDIN(cv,'')) - bag_to_set(USEDIN(cv,
          'AUTOMOTIVE_DESIGN.INTERSECTION_CURVE.BASIS_CURVE')))) = 0)
        THEN
         RETURN (TRUE);
       END_IF;
      ELSE
       IF SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE',
          'AUTOMOTIVE_DESIGN.SURFACE_CURVE'] * TYPEOF(cv)) = 1
        THEN
         IF gbsf_check_curve(cv\surface_curve.curve_3d)
          THEN
           REPEAT i := 1 TO SIZEOF(cv\surface_curve.
                  associated_geometry);
            IF 'AUTOMOTIVE_DESIGN.SURFACE' IN TYPEOF(cv\surface_curve.
               associated_geometry[i])
             THEN
              IF NOT gbsf_check_surface(cv\surface_curve.
                 associated_geometry[i])
               THEN
                RETURN (FALSE);
              END_IF;
             ELSE
              IF 'AUTOMOTIVE_DESIGN.PCURVE' IN TYPEOF(cv\surface_curve
                 .associated_geometry[i])
               THEN
                IF NOT gbsf_check_curve(cv\surface_curve.
                   associated_geometry[i])
                 THEN
                  RETURN (FALSE);
                END_IF;
              END_IF;
            END_IF;
           END_REPEAT;
           RETURN (TRUE);
```

```
                END_IF;
               ELSE
                IF ('AUTOMOTIVE_DESIGN.TRIMMED_CURVE') IN TYPEOF(cv)
                 THEN
                  IF SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE',
                     'AUTOMOTIVE_DESIGN.LINE','AUTOMOTIVE_DESIGN.PARABOLA',
                     'AUTOMOTIVE_DESIGN.HYPERBOLA'] * TYPEOF(cv\trimmed_curve
                     .basis_curve)) = 1
                   THEN
                    RETURN (TRUE);
                   ELSE
                    RETURN (gbsf_check_curve(cv\trimmed_curve.basis_curve));
                  END_IF;
                END_IF;
              END_IF;
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN (FALSE);
END_FUNCTION;


FUNCTION gbsf_check_point(pnt : point) : BOOLEAN;
 IF 'AUTOMOTIVE_DESIGN.CARTESIAN_POINT' IN TYPEOF(pnt)
  THEN
   RETURN (TRUE);
  ELSE
   IF 'AUTOMOTIVE_DESIGN.POINT_ON_CURVE' IN TYPEOF(pnt)
    THEN
     RETURN (gbsf_check_curve(pnt\point_on_curve.basis_curve));
    ELSE
     IF 'AUTOMOTIVE_DESIGN.POINT_ON_SURFACE' IN TYPEOF(pnt)
      THEN
       RETURN (gbsf_check_surface(pnt\point_on_surface.basis_surface));
      ELSE
       IF 'AUTOMOTIVE_DESIGN.DEGENERATE_PCURVE' IN TYPEOF(pnt)
        THEN
         RETURN ((gbsf_check_curve(pnt\degenerate_pcurve.reference_to_curve.
          items[1])) AND (gbsf_check_surface(pnt\degenerate_pcurve.
          basis_surface)));
       END_IF;
     END_IF;
   END_IF;
 END_IF;
 RETURN (FALSE);
END_FUNCTION;


FUNCTION gbsf_check_surface(sf : surface) : BOOLEAN;
```

```
IF (('AUTOMOTIVE_DESIGN.B_SPLINE_SURFACE' IN TYPEOF(sf)) AND (sf\
   b_spline_surface.self_intersect = FALSE))
 THEN
  RETURN (TRUE);
 ELSE
  IF SIZEOF(['AUTOMOTIVE_DESIGN.SPHERICAL_SURFACE',
     'AUTOMOTIVE_DESIGN.TOROIDAL_SURFACE'] * TYPEOF(sf)) = 1
   THEN
    RETURN (TRUE);
   ELSE
    IF 'AUTOMOTIVE_DESIGN.CURVE_BOUNDED_SURFACE' IN TYPEOF(sf)
     THEN
      IF SIZEOF(['AUTOMOTIVE_DESIGN.CONICAL_SURFACE',
         'AUTOMOTIVE_DESIGN.CYLINDRICAL_SURFACE','AUTOMOTIVE_DESIGN.PLANE'] *
          TYPEOF(sf\curve_bounded_surface.basis_surface)) = 1
       THEN
        RETURN (SIZEOF(QUERY(bcurve <* sf\curve_bounded_surface.boundaries |
         NOT (gbsf_check_curve(bcurve)))) = 0);
       ELSE
        IF gbsf_check_surface(sf\curve_bounded_surface.basis_surface)
         THEN
          RETURN (SIZEOF(QUERY(bcurve <* sf\curve_bounded_surface.boundaries
            | NOT (gbsf_check_curve(bcurve)))) = 0);
        END_IF;
      END_IF;
     ELSE
      IF (('AUTOMOTIVE_DESIGN.OFFSET_SURFACE' IN TYPEOF(sf)) AND (sf\
         offset_surface.self_intersect = FALSE))
       THEN
        RETURN (gbsf_check_surface(sf\offset_surface.basis_surface));
       ELSE
        IF 'AUTOMOTIVE_DESIGN.RECTANGULAR_COMPOSITE_SURFACE' IN TYPEOF(sf)
         THEN
          REPEAT i := 1 TO SIZEOF(sf\rectangular_composite_surface.segments);
           REPEAT j := 1 TO SIZEOF(sf\rectangular_composite_surface.segments[
                  i]);
            IF NOT (gbsf_check_surface(sf\rectangular_composite_surface.
               segments[i][j].parent_surface))
             THEN
              RETURN (FALSE);
            END_IF;
           END_REPEAT;
          END_REPEAT;
          RETURN (TRUE);
         ELSE
          IF 'AUTOMOTIVE_DESIGN.RECTANGULAR_TRIMMED_SURFACE' IN TYPEOF(sf)
           THEN
            IF SIZEOF(['AUTOMOTIVE_DESIGN.CONICAL_SURFACE',
               'AUTOMOTIVE_DESIGN.CYLINDRICAL_SURFACE',
               'AUTOMOTIVE_DESIGN.PLANE'] * TYPEOF(sf\
               rectangular_trimmed_surface.basis_surface)) = 1
             THEN
```

```
                RETURN (TRUE);
               ELSE
                RETURN (gbsf_check_surface(sf\rectangular_trimmed_surface.
                 basis_surface));
              END_IF;
             ELSE
              IF 'AUTOMOTIVE_DESIGN.SURFACE_REPLICA' IN TYPEOF(sf)
               THEN
                RETURN (gbsf_check_surface(sf\surface_replica.parent_surface));
               ELSE
                IF 'AUTOMOTIVE_DESIGN.SWEPT_SURFACE' IN TYPEOF(sf)
                 THEN
                  RETURN (gbsf_check_curve(sf\swept_surface.swept_curve));
                END_IF;
              END_IF;
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN (FALSE);
END_FUNCTION;


FUNCTION get_basis_surface(c : curve_on_surface) : SET[0:2] OF surface;
 LOCAL
  surfs : SET[0:2] OF surface;
  n : INTEGER;
 END_LOCAL;
 surfs := [];
 IF 'AUTOMOTIVE_DESIGN.PCURVE' IN TYPEOF(c)
  THEN
   surfs := [c\pcurve.basis_surface];
  ELSE
   IF 'AUTOMOTIVE_DESIGN.SURFACE_CURVE' IN TYPEOF(c)
    THEN
     n := SIZEOF(c\surface_curve.associated_geometry);
     REPEAT i := 1 TO n;
      surfs := surfs + associated_surface(c\surface_curve.associated_geometry[
               i]);
     END_REPEAT;
   END_IF;
 END_IF;
 IF 'AUTOMOTIVE_DESIGN.COMPOSITE_CURVE_ON_SURFACE' IN TYPEOF(c)
  THEN
   n := SIZEOF(c\composite_curve_on_surface.segments);
   surfs := get_basis_surface(c\composite_curve_on_surface.segments[1].
            parent_curve);
   IF n > 1
    THEN
     REPEAT i := 2 TO n;
      surfs := surfs * get_basis_surface(c\composite_curve_on_surface.segments
```

```
               [i].parent_curve);
     END_REPEAT;
   END_IF;
 END_IF;
 RETURN (surfs);
END_FUNCTION;


FUNCTION item_in_context(item : representation_item;cntxt :
                         representation_context) : BOOLEAN;
 LOCAL
  i : INTEGER;
  y : BAG OF representation_item;
 END_LOCAL;
 IF SIZEOF(USEDIN(item,'AUTOMOTIVE_DESIGN.REPRESENTATION.ITEMS') * cntxt.
    representations_in_context) > 0
  THEN
   RETURN (TRUE);
  ELSE
   y := QUERY(z <* USEDIN(item,'') | 'AUTOMOTIVE_DESIGN.REPRESENTATION_ITEM'
        IN TYPEOF(z));
   IF SIZEOF(y) > 0
    THEN
     REPEAT i := 1 TO HIINDEX(y);
      IF item_in_context(y[i],cntxt)
       THEN
        RETURN (TRUE);
      END_IF;
     END_REPEAT;
   END_IF;
 END_IF;
 RETURN (FALSE);
END_FUNCTION;


FUNCTION leap_year(year : year_number) : BOOLEAN;
 IF ((((year MOD 4) = 0) AND ((year MOD 100) <> 0)) OR ((year MOD 400) = 0))
  THEN
   RETURN (TRUE);
  ELSE
   RETURN (FALSE);
 END_IF;
END_FUNCTION;


FUNCTION list_face_loops(f : face) : LIST[0:?] OF loop;
 LOCAL
  loops : LIST[0:?] OF loop := [];
 END_LOCAL;
 REPEAT i := 1 TO SIZEOF(f.bounds);
  loops := loops + (f.bounds[i].bound);
 END_REPEAT;
 RETURN (loops);
```

```
END_FUNCTION;


FUNCTION list_of_topology_reversed(a_list : list_of_reversible_topology_item) :
                                   list_of_reversible_topology_item;
 LOCAL
  the_reverse : list_of_reversible_topology_item;
 END_LOCAL;
 the_reverse := [];
 REPEAT i := 1 TO SIZEOF(a_list);
  the_reverse := topology_reversed(a_list[i]) + the_reverse;
 END_REPEAT;
 RETURN (the_reverse);
END_FUNCTION;


FUNCTION list_to_array(lis : LIST[0:?] OF GENERIC : t;low,u : INTEGER) : ARRAY
                       [low:u] OF GENERIC : t;
 LOCAL
  n : INTEGER;
  res : ARRAY[low:u] OF GENERIC : t;
 END_LOCAL;
 n := SIZEOF(lis);
 IF (n <> (u - low + 1))
  THEN
   RETURN (?);
  ELSE
   REPEAT i := 1 TO n;
    res[low + i - 1] := lis[i];
   END_REPEAT;
   RETURN (res);
 END_IF;
END_FUNCTION;


FUNCTION list_to_set(l : LIST[0:?] OF GENERIC : t) : SET OF GENERIC : t;
 LOCAL
  s : SET OF GENERIC : t := [];
 END_LOCAL;
 REPEAT i := 1 TO SIZEOF(l);
  s := s + l[i];
 END_REPEAT;
 RETURN (s);
END_FUNCTION;


FUNCTION make_array_of_array(lis : LIST[1:?] OF LIST[1:?] OF GENERIC : t;low1,
                             u1,low2,u2 : INTEGER) : ARRAY[low1:u1] OF ARRAY[
                             low2:u2] OF GENERIC : t;
 LOCAL
  n1,n2 : INTEGER;
  res : ARRAY[low1:u1] OF ARRAY[low2:u2] OF GENERIC : t;
  resl : LIST[1:?] OF ARRAY[low2:u2] OF GENERIC : t;
 END_LOCAL;
```

```
 n1 := SIZEOF(lis);
 n2 := SIZEOF(lis[1]);
 IF (n1 <> (u1 - low1 + 1)) AND (n2 <> (u2 - low2 + 1))
  THEN
   RETURN (?);
 END_IF;
 REPEAT i := 1 TO n1;
  IF (SIZEOF(lis[i]) <> n2)
   THEN
    RETURN (?);
  END_IF;
 END_REPEAT;
 REPEAT i := 1 TO n1;
  resl[i] := list_to_array(lis[i],low2,u2);
 END_REPEAT;
 res := list_to_array(resl,low1,u1);
 RETURN (res);
END_FUNCTION;


FUNCTION mixed_loop_type_set(l : SET[0:?] OF loop) : LOGICAL;
 LOCAL
  i : INTEGER;
  poly_loop_type : LOGICAL;
 END_LOCAL;
 IF (SIZEOF(l) <= 1)
  THEN
   RETURN (FALSE);
 END_IF;
 poly_loop_type := ('AUTOMOTIVE_DESIGN.POLY_LOOP' IN TYPEOF(l[1]));
 REPEAT i := 2 TO SIZEOF(l);
  IF (('AUTOMOTIVE_DESIGN.POLY_LOOP' IN TYPEOF(l[i])) <> poly_loop_type)
   THEN
    RETURN (TRUE);
  END_IF;
 END_REPEAT;
 RETURN (FALSE);
END_FUNCTION;


FUNCTION msb_shells(brep : manifold_solid_brep) : SET[1:?] OF closed_shell;
 IF SIZEOF(QUERY(msbtype <* TYPEOF(brep) | msbtype LIKE '*BREP_WITH_VOIDS'))
    >= 1
  THEN
   RETURN (brep\brep_with_voids.voids + brep.outer);
  ELSE
   RETURN ([brep.outer]);
 END_IF;
END_FUNCTION;


FUNCTION nmsf_curve_check(cv : curve) : BOOLEAN;
 IF (('AUTOMOTIVE_DESIGN.B_SPLINE_CURVE' IN TYPEOF(cv)) AND (cv\b_spline_curve
```

```
    .self_intersect = FALSE))
  THEN
   RETURN (TRUE);
  ELSE
   IF SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE','AUTOMOTIVE_DESIGN.CONIC',
      'AUTOMOTIVE_DESIGN.LINE'] * TYPEOF(cv)) = 1
    THEN
     RETURN (TRUE);
    ELSE
     IF SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE',
        'AUTOMOTIVE_DESIGN.CURVE_REPLICA'] * TYPEOF(cv)) = 1
      THEN
       RETURN (nmsf_curve_check(cv\curve_replica.parent_curve));
      ELSE
       IF ((SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE',
          'AUTOMOTIVE_DESIGN.OFFSET_CURVE_3D'] * TYPEOF(cv)) = 1) AND (cv\
          offset_curve_3d.self_intersect = FALSE))
        THEN
         RETURN (nmsf_curve_check(cv\offset_curve_3d.basis_curve));
        ELSE
         IF SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE',
            'AUTOMOTIVE_DESIGN.PCURVE'] * TYPEOF(cv)) = 1
          THEN
           RETURN ((nmsf_curve_check(cv\pcurve.reference_to_curve.items[1]))
            AND (nmsf_surface_check(cv\pcurve.basis_surface)));
          ELSE
           IF SIZEOF(['AUTOMOTIVE_DESIGN.BOUNDED_CURVE',
              'AUTOMOTIVE_DESIGN.SURFACE_CURVE'] * TYPEOF(cv)) = 1
            THEN
             IF nmsf_curve_check(cv\surface_curve.curve_3d)
              THEN
               REPEAT i := 1 TO SIZEOF(cv\surface_curve.associated_geometry);
                IF 'AUTOMOTIVE_DESIGN.SURFACE' IN TYPEOF(cv\surface_curve.
                   associated_geometry[i])
                 THEN
                  IF NOT nmsf_surface_check(cv\surface_curve.
                     associated_geometry[i])
                   THEN
                    RETURN (FALSE);
                  END_IF;
                 ELSE
                  IF 'AUTOMOTIVE_DESIGN.PCURVE' IN TYPEOF(cv\surface_curve.
                     associated_geometry[i])
                   THEN
                    IF NOT nmsf_curve_check(cv\surface_curve.
                       associated_geometry[i])
                     THEN
                      RETURN (FALSE);
                    END_IF;
                  END_IF;
                END_IF;
               END_REPEAT;
```

```
              RETURN (TRUE);
            END_IF;
           ELSE
            IF 'AUTOMOTIVE_DESIGN.POLYLINE' IN TYPEOF(cv)
             THEN
              IF (SIZEOF(cv\polyline.points) >= 3)
               THEN
                RETURN (TRUE);
              END_IF;
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN (FALSE);
END_FUNCTION;


FUNCTION nmsf_surface_check(surf : surface) : BOOLEAN;
 IF 'AUTOMOTIVE_DESIGN.ELEMENTARY_SURFACE' IN TYPEOF(surf)
  THEN
   RETURN (TRUE);
  ELSE
   IF 'AUTOMOTIVE_DESIGN.SWEPT_SURFACE' IN TYPEOF(surf)
    THEN
     RETURN (nmsf_curve_check(surf\swept_surface.swept_curve));
    ELSE
     IF (('AUTOMOTIVE_DESIGN.OFFSET_SURFACE' IN TYPEOF(surf)) AND (surf\
        offset_surface.self_intersect = FALSE))
      THEN
       RETURN (nmsf_surface_check(surf\offset_surface.basis_surface));
      ELSE
       IF 'AUTOMOTIVE_DESIGN.SURFACE_REPLICA' IN TYPEOF(surf)
        THEN
         RETURN (nmsf_surface_check(surf\surface_replica.parent_surface));
        ELSE
         IF (('AUTOMOTIVE_DESIGN.B_SPLINE_SURFACE' IN TYPEOF(surf)) AND (surf\
            b_spline_surface.self_intersect = FALSE))
          THEN
           RETURN (TRUE);
         END_IF;
       END_IF;
     END_IF;
   END_IF;
 END_IF;
 RETURN (FALSE);
END_FUNCTION;


FUNCTION normalise(arg : vector_or_direction) : vector_or_direction;
 LOCAL
```

```
   ndim : INTEGER;
   v : direction;
   result : vector_or_direction;
   vec : vector;
   mag : REAL;
  END_LOCAL;
  IF NOT EXISTS(arg)
   THEN
    result := ?;
   ELSE
    ndim := arg.dim;
    IF 'AUTOMOTIVE_DESIGN.VECTOR' IN TYPEOF(arg)
     THEN
      BEGIN
       vec := arg;
       v := arg.orientation;
       IF arg.magnitude = 0.0
        THEN
         RETURN (?);
        ELSE
         vec.magnitude := 1.0;
       END_IF;
      END;
     ELSE
      v := arg;
    END_IF;
    mag := 0.0;
    REPEAT i := 1 TO ndim;
     mag := mag + v.direction_ratios[i] * v.direction_ratios[i];
    END_REPEAT;
    IF mag > 0.0
     THEN
      mag := SQRT(mag);
      REPEAT i := 1 TO ndim;
       v.direction_ratios[i] := v.direction_ratios[i] / mag;
      END_REPEAT;
      IF 'AUTOMOTIVE_DESIGN.VECTOR' IN TYPEOF(arg)
       THEN
        vec.orientation := v;
        result := vec;
       ELSE
        result := v;
      END_IF;
     ELSE
      RETURN (?);
    END_IF;
  END_IF;
  RETURN (result);
 END_FUNCTION;


 FUNCTION orthogonal_complement(vec : direction) : direction;
  LOCAL
```

```
  result : direction;
 END_LOCAL;
 IF (vec.dim <> 2) OR NOT EXISTS(vec)
  THEN
   RETURN (?);
  ELSE
   result.direction_ratios[1] := - vec.direction_ratios[2];
   result.direction_ratios[2] := vec.direction_ratios[1];
   RETURN (result);
 END_IF;
END_FUNCTION;


FUNCTION path_head_to_tail(a_path : path) : LOGICAL;
 LOCAL
  n : INTEGER;
  p : BOOLEAN := TRUE;
 END_LOCAL;
 n := SIZEOF(a_path.edge_list);
 REPEAT i := 2 TO n;
  p := p AND (a_path.edge_list[i - 1].edge_end :=: a_path.edge_list[i].
       edge_start);
 END_REPEAT;
 RETURN (p);
END_FUNCTION;


FUNCTION path_reversed(a_path : path) : path;
 LOCAL
  the_reverse : path;
 END_LOCAL;
 IF ('AUTOMOTIVE_DESIGN.ORIENTED_PATH' IN TYPEOF(a_path))
  THEN
   the_reverse := oriented_path(a_path\oriented_path.path_element,(NOT (a_path
                  \oriented_path.orientation)));
  ELSE
   the_reverse := oriented_path(a_path,FALSE);
 END_IF;
 RETURN (the_reverse);
END_FUNCTION;


FUNCTION plane_angle_for_pair_in_radian(pair : kinematic_pair;angle :
                                        plane_angle_measure) :
                                        plane_angle_measure;
 LOCAL
  converted_angle : plane_angle_measure := angle;
  link_rep : kinematic_link_representation := representation_of_link(pair.
  joint.first_link);
  link_cntxt : representation_context;
  pa_units : SET OF unit := [];
  pau : unit;
 END_LOCAL;
```

```
link_cntxt := link_rep\representation.context_of_items;
IF NOT ('AUTOMOTIVE_DESIGN.GLOBAL_UNIT_ASSIGNED_CONTEXT' IN TYPEOF(link_cntxt
   ))
 THEN
  RETURN (?);
END_IF;
pa_units := QUERY(unit <* link_cntxt\global_unit_assigned_context.units |
            'AUTOMOTIVE_DESIGN.PLANE_ANGLE_UNIT' IN TYPEOF(unit));
IF SIZEOF(pa_units) <> 1
 THEN
  RETURN (?);
END_IF;
pau := pa_units[1];
IF (NOT ('AUTOMOTIVE_DESIGN.SI_UNIT' IN TYPEOF(pau)) AND NOT (
   'AUTOMOTIVE_DESIGN.CONVERSION_BASED_UNIT' IN TYPEOF(pau)))
 THEN
  RETURN (?);
END_IF;
REPEAT WHILE ('AUTOMOTIVE_DESIGN.CONVERSION_BASED_UNIT' IN TYPEOF(pau));
 converted_angle := converted_angle * pau\conversion_based_unit.
                    conversion_factor.value_component;
 pau := pau\conversion_based_unit.conversion_factor.unit_component;
 IF ((NOT ('AUTOMOTIVE_DESIGN.SI_UNIT' IN TYPEOF(pau)) AND NOT (
    'AUTOMOTIVE_DESIGN.CONVERSION_BASED_UNIT' IN TYPEOF(pau))) OR (NOT (
    'AUTOMOTIVE_DESIGN.PLANE_ANGLE_UNIT' IN TYPEOF(pau))))
  THEN
   RETURN (?);
 END_IF;
END_REPEAT;
IF (pau\si_unit.name <> si_unit_name.radian)
 THEN
  RETURN (?);
END_IF;
CASE pau\si_unit.prefix OF
 si_prefix.exa : RETURN (1.E18 * converted_angle);
 si_prefix.peta : RETURN (1.E15 * converted_angle);
 si_prefix.tera : RETURN (1.E12 * converted_angle);
 si_prefix.giga : RETURN (1.E9 * converted_angle);
 si_prefix.mega : RETURN (1.E6 * converted_angle);
 si_prefix.kilo : RETURN (1.E3 * converted_angle);
 si_prefix.hecto : RETURN (1.E2 * converted_angle);
 si_prefix.deca : RETURN (1.E1 * converted_angle);
 si_prefix.deci : RETURN (1.E-1 * converted_angle);
 si_prefix.centi : RETURN (1.E-2 * converted_angle);
 si_prefix.milli : RETURN (1.E-3 * converted_angle);
 si_prefix.micro : RETURN (1.E-6 * converted_angle);
 si_prefix.nano : RETURN (1.E-9 * converted_angle);
 si_prefix.pico : RETURN (1.E-12 * converted_angle);
 si_prefix.femto : RETURN (1.E-15 * converted_angle);
 si_prefix.atto : RETURN (1.E-18 * converted_angle);
 OTHERWISE : RETURN (converted_angle);
END_CASE;
```

```
END_FUNCTION;


FUNCTION product_class_hierarchy(pc1 : product_class;pc : SET OF product_class
                                 ) : SET OF product_class;
 LOCAL
  i : INTEGER;
  j : INTEGER;
  pc_local : SET OF product_class;
  pcr : SET OF product_concept_relationship;
  temp : SET OF product_class;
  new : BOOLEAN;
 END_LOCAL;
 pc_local := pc;
 pcr := USEDIN(pc1,'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_RELATIONSHIP.' +
        'RELATING_PRODUCT_CONCEPT');
 pcr := pcr + USEDIN(pc1,'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_RELATIONSHIP.' +
        'RELATED_PRODUCT_CONCEPT');
 IF SIZEOF(pcr) = 0
  THEN
   RETURN (pc_local);
 END_IF;
 REPEAT i := 1 TO HIINDEX(pcr);
  IF pcr[i].name = 'class hierarchy'
   THEN
    new := TRUE;
    REPEAT j := 1 TO HIINDEX(pc_local);
     IF pcr[i].relating_product_concept :=: pc_local[j]
      THEN
       new := FALSE;
     END_IF;
    END_REPEAT;
    IF new
     THEN
      pc_local := pc_local + pcr[i].relating_product_concept;
      temp := product_class_hierarchy(pcr[i].relating_product_concept,pc_local
              );
      IF (SIZEOF(pc_local) < SIZEOF(temp))
       THEN
        pc_local := temp;
      END_IF;
    END_IF;
    REPEAT j := 1 TO HIINDEX(pc_local);
     IF pcr[i].related_product_concept :=: pc_local[j]
      THEN
       new := FALSE;
     END_IF;
    END_REPEAT;
    IF new
     THEN
      pc_local := pc_local + pcr[i].related_product_concept;
      temp := product_class_hierarchy(pcr[i].related_product_concept,pc_local)
              ;
```

```
      IF (SIZEOF(pc_local) < SIZEOF(temp))
       THEN
        pc_local := temp;
      END_IF;
    END_IF;
  END_IF;
 END_REPEAT;
 RETURN (pc_local);
END_FUNCTION;


FUNCTION representation_of_link(link : kinematic_link) :
                                kinematic_link_representation;
 LOCAL
  link_rep_rel : BAG OF kinematic_link_representation_relation;
 END_LOCAL;
 link_rep_rel := USEDIN(link,'AUTOMOTIVE_DESIGN.' +
                 'KINEMATIC_LINK_REPRESENTATION_RELATION.' +
                 'TOPOLOGICAL_ASPECTS');
 IF (SIZEOF(link_rep_rel) = 0)
  THEN
   RETURN (?);
  ELSE
   RETURN (link_rep_rel[1].geometric_aspects);
 END_IF;
END_FUNCTION;


FUNCTION representations_mapped_into(rep : representation) : SET OF
                                     representation;
 LOCAL
  results : SET OF representation := [];
  rm : SET OF representation_map;
  mi : SET OF mapped_item;
  i,j : INTEGER;
 END_LOCAL;
 rm := USEDIN(rep,'AUTOMOTIVE_DESIGN.' +
       'REPRESENTATION_MAP.MAPPED_REPRESENTATION');
 REPEAT i := 1 TO HIINDEX(rm);
  mi := mi + rm[i].map_usage;
 END_REPEAT;
 REPEAT j := 1 TO HIINDEX(mi);
  results := results + USEDIN(mi[j],'AUTOMOTIVE_DESIGN.' +
             'REPRESENTATION.ITEMS');
 END_REPEAT;
 RETURN (results);
END_FUNCTION;


FUNCTION scalar_times_vector(scalar : REAL;vec : vector_or_direction) : vector
                             ;
 LOCAL
  v : direction;
```

```
  mag : REAL;
  result : vector;
 END_LOCAL;
 IF NOT EXISTS(scalar) OR NOT EXISTS(vec)
  THEN
   result := ?;
  ELSE
   IF 'AUTOMOTIVE_DESIGN.VECTOR' IN TYPEOF(vec)
    THEN
     v := vec.orientation;
     mag := scalar * vec.magnitude;
    ELSE
     v := vec;
     mag := scalar;
   END_IF;
   IF (mag < 0.0)
    THEN
     REPEAT i := 1 TO SIZEOF(v.direction_ratios);
      v.direction_ratios[i] := - v.direction_ratios[i];
     END_REPEAT;
     mag := - mag;
   END_IF;
   result.orientation := normalise(v);
   result.magnitude := mag;
 END_IF;
 RETURN (result);
END_FUNCTION;


FUNCTION second_proj_axis(z_axis,x_axis,arg : direction) : direction;
 LOCAL
  y_axis : vector;
  v : direction;
  temp : vector;
 END_LOCAL;
 IF NOT EXISTS(arg)
  THEN
   v := direction([0.0,1.0,0.0]);
  ELSE
   v := arg;
 END_IF;
 temp := scalar_times_vector(dot_product(v,z_axis),z_axis);
 y_axis := vector_difference(v,temp);
 temp := scalar_times_vector(dot_product(v,x_axis),x_axis);
 y_axis := vector_difference(y_axis,temp);
 y_axis := normalise(y_axis);
 RETURN (y_axis.orientation);
END_FUNCTION;


FUNCTION set_of_topology_reversed(a_set : set_of_reversible_topology_item) :
                                  set_of_reversible_topology_item;
 LOCAL
```

```
  the_reverse : set_of_reversible_topology_item;
 END_LOCAL;
 the_reverse := [];
 REPEAT i := 1 TO SIZEOF(a_set);
  the_reverse := the_reverse + topology_reversed(a_set[i]);
 END_REPEAT;
 RETURN (the_reverse);
END_FUNCTION;


FUNCTION shell_reversed(a_shell : shell) : shell;
 LOCAL
  the_reverse : shell;
 END_LOCAL;
 IF ('AUTOMOTIVE_DESIGN.ORIENTED_OPEN_SHELL' IN TYPEOF(a_shell))
  THEN
   the_reverse := oriented_open_shell(a_shell\oriented_open_shell.
                  open_shell_element,(NOT (a_shell\oriented_open_shell.
                  orientation)));
  ELSE
   IF ('AUTOMOTIVE_DESIGN.OPEN_SHELL' IN TYPEOF(a_shell))
    THEN
     the_reverse := oriented_open_shell(a_shell,FALSE);
    ELSE
     IF ('AUTOMOTIVE_DESIGN.ORIENTED_CLOSED_SHELL' IN TYPEOF(a_shell))
      THEN
       the_reverse := oriented_closed_shell(a_shell\oriented_closed_shell.
                      closed_shell_element,NOT (a_shell\oriented_closed_shell.
                      orientation));
      ELSE
       IF ('AUTOMOTIVE_DESIGN.CLOSED_SHELL' IN TYPEOF(a_shell))
        THEN
         the_reverse := oriented_closed_shell(a_shell,FALSE);
        ELSE
         the_reverse := ?;
       END_IF;
     END_IF;
   END_IF;
 END_IF;
 RETURN (the_reverse);
END_FUNCTION;


FUNCTION suitably_based_mechanism(mbp : mechanism_base_placement;mech :
                                  mechanism) : BOOLEAN;
 LOCAL
  kprop : kinematic_property_definition;
  kgrep : kinematic_ground_representation;
  klrep : kinematic_link_representation;
  klnk : kinematic_link;
  kjnts : BAG OF kinematic_joint;
  nmechs : BAG OF mechanism;
  nmbps : BAG OF mechanism_base_placement;
```

```
 END_LOCAL;
 kprop := mech.containing_property;
 IF ('AUTOMOTIVE_DESIGN.KINEMATIC_GROUND_REPRESENTATION' IN TYPEOF(mbp\
    representation_relationship.rep_1))
  THEN
   kgrep := mbp\representation_relationship.rep_1;
   IF (kgrep.property\property_definition_representation.definition :=: kprop)
    THEN
     RETURN (TRUE);
    ELSE
     RETURN (FALSE);
   END_IF;
  ELSE
   klrep := mbp\representation_relationship.rep_1;
   klnk := klrep.link_representation_relation.topological_aspects;
   kjnts := USEDIN(klnk,'AUTOMOTIVE_DESIGN.KINEMATIC_JOINT.FIRST_LINK') +
            USEDIN(klnk,'AUTOMOTIVE_DESIGN.KINEMATIC_JOINT.SECOND_LINK');
   nmechs := USEDIN(kjnts[1].structure,
             'AUTOMOTIVE_DESIGN.MECHANISM.STRUCTURE_DEFINITION');
   IF (nmechs[1] :=: mech)
    THEN
     RETURN (FALSE);
    ELSE
     IF (nmechs[1].containing_property :<>: kprop)
      THEN
       RETURN (FALSE);
      ELSE
       nmbps := USEDIN(nmechs[1],'AUTOMOTIVE_DESIGN.' +
                'MECHANISM_BASE_PLACEMENT.BASE_OF_MECHANISM');
       IF (SIZEOF(nmbps) = 0)
        THEN
         RETURN (FALSE);
        ELSE
         RETURN (suitably_based_mechanism(nmbps[1],mech));
       END_IF;
     END_IF;
   END_IF;
 END_IF;
END_FUNCTION;


FUNCTION surface_weights_positive(b : rational_b_spline_surface) : BOOLEAN;
 LOCAL
  result : BOOLEAN := TRUE;
 END_LOCAL;
 REPEAT i := 0 TO b.u_upper;
  REPEAT j := 0 TO b.v_upper;
   IF (b.weights[i][j] <= 0.0)
    THEN
     result := FALSE;
     RETURN (result);
   END_IF;
  END_REPEAT;
```

```
  END_REPEAT;
  RETURN (result);
 END_FUNCTION;


 FUNCTION topology_reversed(an_item : reversible_topology) : reversible_topology
                            ;
  IF ('AUTOMOTIVE_DESIGN.EDGE' IN TYPEOF(an_item))
   THEN
    RETURN (edge_reversed(an_item));
  END_IF;
  IF ('AUTOMOTIVE_DESIGN.PATH' IN TYPEOF(an_item))
   THEN
    RETURN (path_reversed(an_item));
  END_IF;
  IF ('AUTOMOTIVE_DESIGN.FACE_BOUND' IN TYPEOF(an_item))
   THEN
    RETURN (face_bound_reversed(an_item));
  END_IF;
  IF ('AUTOMOTIVE_DESIGN.FACE' IN TYPEOF(an_item))
   THEN
    RETURN (face_reversed(an_item));
  END_IF;
  IF ('AUTOMOTIVE_DESIGN.SHELL' IN TYPEOF(an_item))
   THEN
    RETURN (shell_reversed(an_item));
  END_IF;
  IF ('SET' IN TYPEOF(an_item))
   THEN
    RETURN (set_of_topology_reversed(an_item));
  END_IF;
  IF ('LIST' IN TYPEOF(an_item))
   THEN
    RETURN (list_of_topology_reversed(an_item));
  END_IF;
  RETURN (?);
 END_FUNCTION;


 FUNCTION unique_link_usage(link : kinematic_link) : BOOLEAN;
  LOCAL
   mechs : SET OF mechanism;
   joints : SET OF kinematic_joint;
   struct : kinematic_structure;
  END_LOCAL;
  joints := bag_to_set(USEDIN(link,
            'AUTOMOTIVE_DESIGN.KINEMATIC_JOINT.FIRST_LINK') + USEDIN(link,
            'AUTOMOTIVE_DESIGN.KINEMATIC_JOINT.SECOND_LINK'));
  struct := joints[1].structure;
  REPEAT i := 2 TO SIZEOF(joints);
   IF (joints[i].structure :<>: struct)
    THEN
     RETURN (FALSE);
```

```
  END_IF;
 END_REPEAT;
 mechs := bag_to_set(USEDIN(struct,
          'AUTOMOTIVE_DESIGN.MECHANISM.STRUCTURE_DEFINITION'));
 IF (SIZEOF(mechs) <> 1)
  THEN
   RETURN (FALSE);
 END_IF;
 RETURN (TRUE);
END_FUNCTION;


FUNCTION using_representations(item : representation_item) : SET OF
                               representation;
 LOCAL
  results : SET OF representation;
  result_bag : BAG OF representation;
  intermediate_items : SET OF representation_item;
  i : INTEGER;
 END_LOCAL;
 result_bag := USEDIN(item,'AUTOMOTIVE_DESIGN.REPRESENTATION.ITEMS');
 IF SIZEOF(result_bag) > 0
  THEN
   REPEAT i := 1 TO HIINDEX(result_bag);
    results := results + result_bag[i];
   END_REPEAT;
 END_IF;
 intermediate_items := QUERY(z <* bag_to_set(USEDIN(item,'')) |
                       'AUTOMOTIVE_DESIGN.REPRESENTATION_ITEM' IN TYPEOF(z));
 IF SIZEOF(intermediate_items) > 0
  THEN
   REPEAT i := 1 TO HIINDEX(intermediate_items);
    results := results + using_representations(intermediate_items[i]);
   END_REPEAT;
 END_IF;
 RETURN (results);
END_FUNCTION;


FUNCTION valid_basis_curve_in_2d_wireframe(crv : curve;schma : STRING) :
                                           BOOLEAN;
 IF (schma + '.TRIMMED_CURVE') IN TYPEOF(crv)
  THEN
   RETURN (valid_basis_curve_in_2d_wireframe(crv\trimmed_curve.basis_curve,
    schma));
  ELSE
   IF (schma + '.COMPOSITE_CURVE') IN TYPEOF(crv)
    THEN
     RETURN (SIZEOF(QUERY(ccs <* crv\composite_curve.segments | NOT (
      valid_basis_curve_in_2d_wireframe(ccs.parent_curve,schma)))) = 0);
    ELSE
     IF ('AUTOMOTIVE_DESIGN.OFFSET_CURVE_2D') IN TYPEOF(crv)
      THEN
```

```
        RETURN (valid_basis_curve_in_2d_wireframe(crv\offset_curve_2d.
         basis_curve,schma));
       ELSE
        IF SIZEOF(['AUTOMOTIVE_DESIGN.LINE','AUTOMOTIVE_DESIGN.B_SPLINE_CURVE',
           'AUTOMOTIVE_DESIGN.CIRCLE','AUTOMOTIVE_DESIGN.ELLIPSE',
           'AUTOMOTIVE_DESIGN.HYPERBOLA','AUTOMOTIVE_DESIGN.PARABOLA',
           'AUTOMOTIVE_DESIGN.POLYLINE'] * TYPEOF(crv)) = 1
         THEN
          RETURN (TRUE);
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN (FALSE);
END_FUNCTION;


FUNCTION valid_calendar_date(date : calendar_date) : LOGICAL;
 IF NOT ({1<=date.day_component<=31})
  THEN
   RETURN (FALSE);
 END_IF;
 CASE date.month_component OF
  4 : RETURN ({1<=date.day_component<=30});
  6 : RETURN ({1<=date.day_component<=30});
  9 : RETURN ({1<=date.day_component<=30});
  11 : RETURN ({1<=date.day_component<=30});
  2 : BEGIN
       IF (leap_year(date.year_component))
        THEN
         RETURN ({1<=date.day_component<=29});
        ELSE
         RETURN ({1<=date.day_component<=28});
       END_IF;
      END;
  OTHERWISE : RETURN (TRUE);
 END_CASE;
END_FUNCTION;


FUNCTION valid_geometrically_bounded_wf_curve(crv : curve) : BOOLEAN;
 IF SIZEOF(['AUTOMOTIVE_DESIGN.POLYLINE','AUTOMOTIVE_DESIGN.B_SPLINE_CURVE',
    'AUTOMOTIVE_DESIGN.ELLIPSE','AUTOMOTIVE_DESIGN.CIRCLE'] * TYPEOF(crv)) = 1
  THEN
   RETURN (TRUE);
  ELSE
   IF (('AUTOMOTIVE_DESIGN.TRIMMED_CURVE') IN TYPEOF(crv))
    THEN
     IF SIZEOF(['AUTOMOTIVE_DESIGN.LINE','AUTOMOTIVE_DESIGN.PARABOLA',
        'AUTOMOTIVE_DESIGN.HYPERBOLA'] * TYPEOF(crv\trimmed_curve.basis_curve)
        ) = 1
      THEN
       RETURN (TRUE);
```

```
      ELSE
       RETURN (valid_geometrically_bounded_wf_curve(crv\trimmed_curve.
        basis_curve));
     END_IF;
    ELSE
     IF (('AUTOMOTIVE_DESIGN.OFFSET_CURVE_3D') IN TYPEOF(crv))
      THEN
       RETURN (valid_geometrically_bounded_wf_curve(crv\offset_curve_3d.
        basis_curve));
      ELSE
       IF (('AUTOMOTIVE_DESIGN.CURVE_REPLICA') IN TYPEOF(crv))
        THEN
         RETURN (valid_geometrically_bounded_wf_curve(crv\curve_replica.
          parent_curve));
        ELSE
         IF (('AUTOMOTIVE_DESIGN.COMPOSITE_CURVE') IN TYPEOF(crv))
          THEN
           RETURN (SIZEOF(QUERY(ccs <* crv\composite_curve.segments | NOT (
            valid_geometrically_bounded_wf_curve(ccs.parent_curve)))) = 0);
         END_IF;
       END_IF;
     END_IF;
   END_IF;
 END_IF;
 RETURN (FALSE);
END_FUNCTION;


FUNCTION valid_geometrically_bounded_wf_point(pnt : point) : BOOLEAN;
 IF (('AUTOMOTIVE_DESIGN.CARTESIAN_POINT') IN TYPEOF(pnt))
  THEN
   RETURN (TRUE);
  ELSE
   IF (('AUTOMOTIVE_DESIGN.POINT_ON_CURVE') IN TYPEOF(pnt))
    THEN
     RETURN (valid_geometrically_bounded_wf_curve(pnt\point_on_curve.
      basis_curve));
    ELSE
     IF (('AUTOMOTIVE_DESIGN.POINT_REPLICA') IN TYPEOF(pnt))
      THEN
       RETURN (valid_geometrically_bounded_wf_point(pnt\point_replica.
        parent_pt));
     END_IF;
   END_IF;
 END_IF;
 RETURN (FALSE);
END_FUNCTION;


FUNCTION valid_time(time : local_time) : BOOLEAN;
 IF EXISTS(time.second_component)
  THEN
   RETURN (EXISTS(time.minute_component));
```

```
   ELSE
    RETURN (TRUE);
  END_IF;
 END_FUNCTION;


 FUNCTION valid_units(m : measure_with_unit) : BOOLEAN;
  IF 'AUTOMOTIVE_DESIGN.LENGTH_MEASURE' IN TYPEOF(m.value_component)
   THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
       1.0,0.0,0.0,0.0,0.0,0.0,0.0)
     THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'AUTOMOTIVE_DESIGN.MASS_MEASURE' IN TYPEOF(m.value_component)
   THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
       0.0,1.0,0.0,0.0,0.0,0.0,0.0)
     THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'AUTOMOTIVE_DESIGN.TIME_MEASURE' IN TYPEOF(m.value_component)
   THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
       0.0,0.0,1.0,0.0,0.0,0.0,0.0)
     THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'AUTOMOTIVE_DESIGN.ELECTRIC_CURRENT_MEASURE' IN TYPEOF(m.value_component)
   THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
       0.0,0.0,0.0,1.0,0.0,0.0,0.0)
     THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'AUTOMOTIVE_DESIGN.THERMODYNAMIC_TEMPERATURE_MEASURE' IN TYPEOF(m.
     value_component)
   THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
       0.0,0.0,0.0,0.0,1.0,0.0,0.0)
     THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'AUTOMOTIVE_DESIGN.AMOUNT_OF_SUBSTANCE_MEASURE' IN TYPEOF(m.
     value_component)
   THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
       0.0,0.0,0.0,0.0,0.0,1.0,0.0)
```

```
   THEN
    RETURN (FALSE);
  END_IF;
END_IF;
IF 'AUTOMOTIVE_DESIGN.LUMINOUS_INTENSITY_MEASURE' IN TYPEOF(m.value_component
   )
 THEN
  IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
     0.0,0.0,0.0,0.0,0.0,0.0,1.0)
   THEN
    RETURN (FALSE);
  END_IF;
END_IF;
IF 'AUTOMOTIVE_DESIGN.PLANE_ANGLE_MEASURE' IN TYPEOF(m.value_component)
 THEN
  IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
     0.0,0.0,0.0,0.0,0.0,0.0,0.0)
   THEN
    RETURN (FALSE);
  END_IF;
END_IF;
IF 'AUTOMOTIVE_DESIGN.SOLID_ANGLE_MEASURE' IN TYPEOF(m.value_component)
 THEN
  IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
     0.0,0.0,0.0,0.0,0.0,0.0,0.0)
   THEN
    RETURN (FALSE);
  END_IF;
END_IF;
IF 'AUTOMOTIVE_DESIGN.AREA_MEASURE' IN TYPEOF(m.value_component)
 THEN
  IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
     2.0,0.0,0.0,0.0,0.0,0.0,0.0)
   THEN
    RETURN (FALSE);
  END_IF;
END_IF;
IF 'AUTOMOTIVE_DESIGN.VOLUME_MEASURE' IN TYPEOF(m.value_component)
 THEN
  IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
     3.0,0.0,0.0,0.0,0.0,0.0,0.0)
   THEN
    RETURN (FALSE);
  END_IF;
END_IF;
IF 'AUTOMOTIVE_DESIGN.RATIO_MEASURE' IN TYPEOF(m.value_component)
 THEN
  IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
     0.0,0.0,0.0,0.0,0.0,0.0,0.0)
   THEN
    RETURN (FALSE);
  END_IF;
```

```
  END_IF;
  IF 'AUTOMOTIVE_DESIGN.POSITIVE_LENGTH_MEASURE' IN TYPEOF(m.value_component)
   THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
       1.0,0.0,0.0,0.0,0.0,0.0,0.0)
     THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  IF 'AUTOMOTIVE_DESIGN.POSITIVE_PLANE_ANGLE_MEASURE' IN TYPEOF(m.
     value_component)
   THEN
    IF derive_dimensional_exponents(m.unit_component) <> dimensional_exponents(
       0.0,0.0,0.0,0.0,0.0,0.0,0.0)
     THEN
      RETURN (FALSE);
    END_IF;
  END_IF;
  RETURN (TRUE);
 END_FUNCTION;


 FUNCTION valid_wireframe_edge_curve(crv : curve) : BOOLEAN;
  IF SIZEOF(['AUTOMOTIVE_DESIGN.LINE','AUTOMOTIVE_DESIGN.CIRCLE',
     'AUTOMOTIVE_DESIGN.ELLIPSE','AUTOMOTIVE_DESIGN.PARABOLA',
     'AUTOMOTIVE_DESIGN.HYPERBOLA','AUTOMOTIVE_DESIGN.B_SPLINE_CURVE',
     'AUTOMOTIVE_DESIGN.POLYLINE'] * TYPEOF(crv)) = 1
   THEN
    RETURN (TRUE);
   ELSE
    IF ('AUTOMOTIVE_DESIGN.CURVE_REPLICA') IN TYPEOF(crv)
     THEN
      RETURN (valid_wireframe_edge_curve(crv\curve_replica.parent_curve));
     ELSE
      IF ('AUTOMOTIVE_DESIGN.OFFSET_CURVE_3D') IN TYPEOF(crv)
       THEN
        RETURN (valid_wireframe_edge_curve(crv\offset_curve_3d.basis_curve));
      END_IF;
    END_IF;
  END_IF;
  RETURN (FALSE);
 END_FUNCTION;


 FUNCTION valid_wireframe_vertex_point(pnt : point) : BOOLEAN;
  IF ('AUTOMOTIVE_DESIGN.CARTESIAN_POINT') IN TYPEOF(pnt)
   THEN
    RETURN (TRUE);
   ELSE
    IF ('AUTOMOTIVE_DESIGN.POINT_REPLICA') IN TYPEOF(pnt)
     THEN
      RETURN (valid_wireframe_vertex_point(pnt\point_replica.parent_pt));
    END_IF;
```

```
 END_IF;
 RETURN (FALSE);
END_FUNCTION;

FUNCTION vector_difference(arg1,arg2 : vector_or_direction) : vector;
 LOCAL
  result : vector;
  res,vec1,vec2 : direction;
  mag,mag1,mag2 : REAL;
  ndim : INTEGER;
 END_LOCAL;
 IF ((NOT EXISTS(arg1)) OR (NOT EXISTS(arg2))) OR (arg1.dim <> arg2.dim)
  THEN
   result := ?;
  ELSE
   BEGIN
    IF 'AUTOMOTIVE_DESIGN.VECTOR' IN TYPEOF(arg1)
     THEN
      mag1 := arg1.magnitude;
      vec1 := arg1.orientation;
     ELSE
      mag1 := 1.0;
      vec1 := arg1;
    END_IF;
    IF 'AUTOMOTIVE_DESIGN.VECTOR' IN TYPEOF(arg2)
     THEN
      mag2 := arg2.magnitude;
      vec2 := arg2.orientation;
     ELSE
      mag2 := 1.0;
      vec2 := arg2;
    END_IF;
    vec1 := normalise(vec1);
    vec2 := normalise(vec2);
    ndim := SIZEOF(vec1.direction_ratios);
    mag := 0.0;
    REPEAT i := 1 TO ndim;
     res.direction_ratios[i] := mag1 * vec1.direction_ratios[i] - mag2 * vec2.
                                direction_ratios[i];
     mag := mag + (res.direction_ratios[i] * res.direction_ratios[i]);
    END_REPEAT;
    IF (mag > 0.0)
     THEN
      result.magnitude := SQRT(mag);
      result.orientation := res;
     ELSE
      result.magnitude := 0.0;
      result.orientation := vec1;
    END_IF;
   END;
 END_IF;
 RETURN (result);
```

```
END_FUNCTION;


FUNCTION ypr_index(ypr : ypr_enumeration) : INTEGER;
 CASE ypr OF
  yaw : RETURN (1);
  pitch : RETURN (2);
  roll : RETURN (3);
 END_CASE;
END_FUNCTION;


RULE action_assignment_subtype_exclusiveness FOR (action_assignment);
 WHERE
  wr1 : SIZEOF(QUERY(aa <* action_assignment | (SIZEOF(TYPEOF(aa) * [
        'AUTOMOTIVE_DESIGN.APPLIED_ACTION_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.ELEMENT_DELIVERY_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.EVENT_CONTEXT_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.PROCESS_PLAN_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.RENTENTION_ASSIGNMENT']) > 1) OR ((SIZEOF(TYPEOF(aa
        )) > 2) AND (SIZEOF(TYPEOF(aa) * [
        'AUTOMOTIVE_DESIGN.APPLIED_ACTION_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.ELEMENT_DELIVERY_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.EVENT_CONTEXT_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.PROCESS_PLAN_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.RENTENTION_ASSIGNMENT']) = 1)))) = 0;
END_RULE;


RULE action_subtype_restriction FOR (action);
 WHERE
  wr1 : SIZEOF(QUERY(a <* action | (SIZEOF(TYPEOF(a) * [
        'AUTOMOTIVE_DESIGN.EXECUTED_ACTION',
        'AUTOMOTIVE_DESIGN.EVENT_REFERENCE','AUTOMOTIVE_DESIGN.RENTENTION']) >
         1) OR ((SIZEOF(TYPEOF(a)) > 2) AND (SIZEOF(TYPEOF(a) * [
        'AUTOMOTIVE_DESIGN.EXECUTED_ACTION',
        'AUTOMOTIVE_DESIGN.EVENT_REFERENCE','AUTOMOTIVE_DESIGN.RENTENTION']) =
         1)))) = 0;
END_RULE;


RULE activity_requires_person_or_organization_assignment FOR (action);
 WHERE
  wr1 : SIZEOF(QUERY(a <* action | (SIZEOF(TYPEOF(a)) = 1) AND (a.description
        <> 'process operation') AND (SIZEOF(QUERY(u <* USEDIN(a,
        'AUTOMOTIVE_DESIGN.' + 'APPLIED_PERSON_ASSIGNMENT.ITEMS') | (u.
        person_role.name = 'custodian'))) + SIZEOF(QUERY(u <* USEDIN(a,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') | (u.
        person_and_organization_role.name = 'custodian'))) + SIZEOF(QUERY(u <*
        USEDIN(a,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | (u.organization_role.name =
```

```
         'custodian'))) = 0))) = 0;
END_RULE;


RULE application_context_requires_ap_definition FOR (application_context);
 WHERE
  wr1 : SIZEOF(QUERY(ac <* application_context | SIZEOF(QUERY(apd <* USEDIN(ac
        ,'AUTOMOTIVE_DESIGN.' + 'APPLICATION_PROTOCOL_DEFINITION.APPLICATION')
         | apd.application_interpreted_model_schema_name = 'automotive_design'
        )) <> 1)) = 0;
END_RULE;


RULE approval_person_organization_requires_date_time FOR (
     approval_person_organization);
 WHERE
  wr1 : SIZEOF(QUERY(apo <* approval_person_organization | (SIZEOF(QUERY(u <*
        USEDIN(apo,'AUTOMOTIVE_DESIGN.' + 'APPLIED_DATE_ASSIGNMENT.ITEMS') | (
        u.date_role.name = 'actual'))) + SIZEOF(QUERY(u <* USEDIN(apo,
        'AUTOMOTIVE_DESIGN.' + 'APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') | (u.
        date_time_role.name = 'actual')))) <> 1)) = 0;
END_RULE;


RULE approval_requires_approval_assignment FOR (approval);
 WHERE
  wr1 : SIZEOF(QUERY(a <* approval | SIZEOF(USEDIN(a,'AUTOMOTIVE_DESIGN.' +
        'APPROVAL_ASSIGNMENT.ASSIGNED_APPROVAL')) = 0)) = 0;
END_RULE;


RULE characterized_object_subtype_exclusiveness FOR (characterized_object);
 WHERE
  wr1 : SIZEOF(QUERY(co <* characterized_object | (SIZEOF(TYPEOF(co) * [
        'AUTOMOTIVE_DESIGN.CHARACTERIZED_PRODUCT_CATEGORY',
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION',
        'AUTOMOTIVE_DESIGN.FEATURE_DEFINITION',
        'AUTOMOTIVE_DESIGN.PRODUCT_CLASS',
        'AUTOMOTIVE_DESIGN.PRODUCT_COMPONENT']) > 1) OR ((SIZEOF(TYPEOF(co)) >
         2) AND (SIZEOF(TYPEOF(co) * [
        'AUTOMOTIVE_DESIGN.CHARACTERIZED_PRODUCT_CATEGORY',
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_DEFINITION',
        'AUTOMOTIVE_DESIGN.FEATURE_DEFINITION',
        'AUTOMOTIVE_DESIGN.PRODUCT_CLASS',
        'AUTOMOTIVE_DESIGN.PRODUCT_COMPONENT']) = 1)))) = 0;
END_RULE;


RULE configuration_design_subtype_exclusiveness FOR (configuration_design);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* configuration_design | (SIZEOF(TYPEOF(pd) * [
        'AUTOMOTIVE_DESIGN.component_function_definition',
        'AUTOMOTIVE_DESIGN.occurrence_usage_definition',
        'AUTOMOTIVE_DESIGN.solution_alternative_definition']) > 1) OR ((SIZEOF
```

```
        (TYPEOF(pd)) > 2) AND (SIZEOF(TYPEOF(pd) * [
        'AUTOMOTIVE_DESIGN.component_function_definition',
        'AUTOMOTIVE_DESIGN.occurrence_usage_definition',
        'AUTOMOTIVE_DESIGN.solution_alternative_definition']) = 1)))) = 0;
END_RULE;


RULE coordinated_assembly_and_shape FOR (next_assembly_usage_occurrence);
 WHERE
  wr1 : SIZEOF(QUERY(nauo <* next_assembly_usage_occurrence | NOT (
        assembly_shape_is_defined(nauo)))) = 0;
END_RULE;


RULE curve_font_usage FOR (curve_style_font,externally_defined_curve_font,
                           pre_defined_curve_font);
 WHERE
  wr1 : SIZEOF(QUERY(csf <* curve_style_font | NOT (SIZEOF(USEDIN(csf,
        'AUTOMOTIVE_DESIGN.' + 'CURVE_STYLE.CURVE_FONT')) >= 1))) = 0;
  wr2 : SIZEOF(QUERY(pdcf <* pre_defined_curve_font | NOT (SIZEOF(USEDIN(pdcf,
        'AUTOMOTIVE_DESIGN.' + 'CURVE_STYLE.CURVE_FONT')) >= 1))) = 0;
  wr3 : SIZEOF(QUERY(edcf <* externally_defined_curve_font | NOT (SIZEOF(
        USEDIN(edcf,'AUTOMOTIVE_DESIGN.' + 'CURVE_STYLE.CURVE_FONT')) >= 1)))
        = 0;
END_RULE;


RULE definition_replacement_requires_effectivity_assignment FOR (
     product_definition_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(pdr <* product_definition_relationship | (pdr.name =
        'definition replacement') AND (SIZEOF(QUERY(aeia <* USEDIN(pdr,
        'AUTOMOTIVE_DESIGN.' + 'APPLIED_EFFECTIVE_ITEM_ASSIGNMENT.ITEMS') | ((
        'AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY' IN TYPEOF(aeia.
        assigned_effectivity)) OR ('AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY'
         IN TYPEOF(aeia.assigned_effectivity))) AND (SIZEOF(QUERY(ea <* USEDIN
        (aeia.assigned_effectivity,'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT.ITEMS')
         | ea.action_role.name = 'effective start')) > 0))) = 0))) = 0;
END_RULE;


RULE dependent_instantiable_action_request_role FOR (action_request_role);
 WHERE
  wr1 : SIZEOF(QUERY(a <* action_request_role | NOT (SIZEOF(USEDIN(a,'')) > 0)
        )) = 0;
END_RULE;


RULE dependent_instantiable_action_role FOR (action_role);
 WHERE
  wr1 : SIZEOF(QUERY(a <* action_role | NOT (SIZEOF(USEDIN(a,'')) > 0))) = 0;
END_RULE;
```

```
RULE dependent_instantiable_approval_date_time_role FOR (
     approval_date_time_role);
 WHERE
  wr1 : SIZEOF(QUERY(a <* approval_date_time_role | NOT (SIZEOF(USEDIN(a,''))
        > 0))) = 0;
END_RULE;


RULE dependent_instantiable_approval_role FOR (approval_role);
 WHERE
  wr1 : SIZEOF(QUERY(a <* approval_role | NOT (SIZEOF(USEDIN(a,'')) > 0))) = 0
        ;
END_RULE;


RULE dependent_instantiable_approver_role FOR (approver_role);
 WHERE
  wr1 : SIZEOF(QUERY(a <* approver_role | NOT (SIZEOF(USEDIN(a,'')) > 0))) = 0
        ;
END_RULE;


RULE dependent_instantiable_certification_role FOR (certification_role);
 WHERE
  wr1 : SIZEOF(QUERY(c <* certification_role | NOT (SIZEOF(USEDIN(c,'')) > 0))
        ) = 0;
END_RULE;


RULE dependent_instantiable_colour_rgb FOR (colour_rgb);
 WHERE
  wr1 : SIZEOF(QUERY(c <* colour_rgb | NOT (SIZEOF(USEDIN(c,'')) > 0))) = 0;
END_RULE;


RULE dependent_instantiable_contract_role FOR (contract_role);
 WHERE
  wr1 : SIZEOF(QUERY(c <* contract_role | NOT (SIZEOF(USEDIN(c,'')) > 0))) = 0
        ;
END_RULE;


RULE dependent_instantiable_curve_style FOR (curve_style);
 WHERE
  wr1 : SIZEOF(QUERY(c <* curve_style | NOT (SIZEOF(USEDIN(c,'')) > 0))) = 0;
END_RULE;


RULE dependent_instantiable_date FOR (date);
 WHERE
  wr1 : SIZEOF(QUERY(d <* date | NOT (SIZEOF(USEDIN(d,'')) > 0))) = 0;
END_RULE;
```

```
RULE dependent_instantiable_date_and_time FOR (date_and_time);
 WHERE
  wr1 : SIZEOF(QUERY(d <* date_and_time | NOT (SIZEOF(USEDIN(d,'')) > 0))) = 0
        ;
END_RULE;


RULE dependent_instantiable_date_role FOR (date_role);
 WHERE
  wr1 : SIZEOF(QUERY(d <* date_role | NOT (SIZEOF(USEDIN(d,'')) > 0))) = 0;
END_RULE;


RULE dependent_instantiable_date_time_role FOR (date_time_role);
 WHERE
  wr1 : SIZEOF(QUERY(d <* date_time_role | NOT (SIZEOF(USEDIN(d,'')) > 0))) =
        0;
END_RULE;


RULE dependent_instantiable_derived_unit FOR (derived_unit);
 WHERE
  wr1 : SIZEOF(QUERY(d <* derived_unit | NOT (SIZEOF(USEDIN(d,'')) > 0))) = 0;
END_RULE;


RULE dependent_instantiable_dimensional_size FOR (dimensional_size);
 WHERE
  wr1 : SIZEOF(QUERY(d <* dimensional_size | NOT (SIZEOF(USEDIN(d,'')) > 0)))
        = 0;
END_RULE;


RULE dependent_instantiable_document_role FOR (document_role);
 WHERE
  wr1 : SIZEOF(QUERY(d <* document_role | NOT (SIZEOF(USEDIN(d,'')) > 0))) = 0
        ;
END_RULE;


RULE dependent_instantiable_effectivity_role FOR (effectivity_role);
 WHERE
  wr1 : SIZEOF(QUERY(e <* effectivity_role | NOT (SIZEOF(USEDIN(e,'')) > 0)))
        = 0;
END_RULE;


RULE dependent_instantiable_externally_defined_symbol FOR (
     externally_defined_symbol);
 WHERE
  wr1 : SIZEOF(QUERY(e <* externally_defined_symbol | NOT (SIZEOF(USEDIN(e,'')
        ) > 0))) = 0;
END_RULE;
```

```
RULE dependent_instantiable_fill_area_style FOR (fill_area_style);
 WHERE
  wr1 : SIZEOF(QUERY(f <* fill_area_style | NOT (SIZEOF(USEDIN(f,'')) > 0))) =
         0;
END_RULE;


RULE dependent_instantiable_fill_area_style_colour FOR (fill_area_style_colour
                                                        );
 WHERE
  wr1 : SIZEOF(QUERY(f <* fill_area_style_colour | NOT (SIZEOF(USEDIN(f,'')) >
         0))) = 0;
END_RULE;


RULE dependent_instantiable_group_role FOR (group_role);
 WHERE
  wr1 : SIZEOF(QUERY(g <* group_role | NOT (SIZEOF(USEDIN(g,'')) > 0))) = 0;
END_RULE;


RULE dependent_instantiable_identification_role FOR (identification_role);
 WHERE
  wr1 : SIZEOF(QUERY(i <* identification_role | NOT (SIZEOF(USEDIN(i,'')) > 0)
        )) = 0;
END_RULE;


RULE dependent_instantiable_language_role FOR (language_role);
 WHERE
  wr1 : SIZEOF(QUERY(l <* language_role | NOT (SIZEOF(USEDIN(l,'')) > 0))) = 0
        ;
END_RULE;


RULE dependent_instantiable_measure_with_unit FOR (measure_with_unit);
 WHERE
  wr1 : SIZEOF(QUERY(m <* measure_with_unit | NOT (SIZEOF(USEDIN(m,'')) > 0)))
         = 0;
END_RULE;


RULE dependent_instantiable_named_unit FOR (named_unit);
 WHERE
  wr1 : SIZEOF(QUERY(n <* named_unit | NOT (SIZEOF(USEDIN(n,'')) > 0))) = 0;
END_RULE;


RULE dependent_instantiable_organization_role FOR (organization_role);
 WHERE
  wr1 : SIZEOF(QUERY(o <* organization_role | NOT (SIZEOF(USEDIN(o,'')) > 0)))
         = 0;
END_RULE;
```

```
RULE dependent_instantiable_person_and_organization_role FOR (
     person_and_organization_role);
 WHERE
  wr1 : SIZEOF(QUERY(p <* person_and_organization_role | NOT (SIZEOF(USEDIN(p,
        '')) > 0))) = 0;
END_RULE;


RULE dependent_instantiable_person_role FOR (person_role);
 WHERE
  wr1 : SIZEOF(QUERY(p <* person_role | NOT (SIZEOF(USEDIN(p,'')) > 0))) = 0;
END_RULE;


RULE dependent_instantiable_pre_defined_colour FOR (pre_defined_colour);
 WHERE
  wr1 : SIZEOF(QUERY(p <* pre_defined_colour | NOT (SIZEOF(USEDIN(p,'')) > 0))
        ) = 0;
END_RULE;


RULE dependent_instantiable_pre_defined_symbol FOR (pre_defined_symbol);
 WHERE
  wr1 : SIZEOF(QUERY(p <* pre_defined_symbol | NOT (SIZEOF(USEDIN(p,'')) > 0))
        ) = 0;
END_RULE;


RULE dependent_instantiable_presentation_style_by_context FOR (
     presentation_style_by_context);
 WHERE
  wr1 : SIZEOF(QUERY(p <* presentation_style_by_context | NOT (SIZEOF(USEDIN(p
        ,'')) > 0))) = 0;
END_RULE;


RULE dependent_instantiable_representation FOR (representation);
 WHERE
  wr1 : SIZEOF(QUERY(r <* representation | NOT (SIZEOF(USEDIN(r,'')) > 0))) =
        0;
END_RULE;


RULE dependent_instantiable_security_classification_role FOR (
     security_classification_role);
 WHERE
  wr1 : SIZEOF(QUERY(s <* security_classification_role | NOT (SIZEOF(USEDIN(s,
        '')) > 0))) = 0;
END_RULE;


RULE dependent_instantiable_symbol_colour FOR (symbol_colour);
 WHERE
  wr1 : SIZEOF(QUERY(s <* symbol_colour | NOT (SIZEOF(USEDIN(s,'')) > 0))) = 0
        ;
```

```
END_RULE;

RULE dependent_instantiable_text_style FOR (text_style);
 WHERE
  wr1 : SIZEOF(QUERY(t <* text_style | NOT (SIZEOF(USEDIN(t,'')) > 0))) = 0;
END_RULE;

RULE dependent_instantiable_text_style_for_defined_font FOR (
     text_style_for_defined_font);
 WHERE
  wr1 : SIZEOF(QUERY(t <* text_style_for_defined_font | NOT (SIZEOF(USEDIN(t,
        '')) > 0))) = 0;
END_RULE;

RULE dependent_instantiable_tolerance_value FOR (tolerance_value);
 WHERE
  wr1 : SIZEOF(QUERY(t <* tolerance_value | NOT (SIZEOF(USEDIN(t,'')) > 0))) =
         0;
END_RULE;

RULE dependent_instantiable_uncertainty_qualifier FOR (uncertainty_qualifier);
 WHERE
  wr1 : SIZEOF(QUERY(u <* uncertainty_qualifier | NOT (SIZEOF(USEDIN(u,'')) >
        0))) = 0;
END_RULE;

RULE dimensionality_is_two_or_three FOR (geometric_representation_context);
 WHERE
  wr1 : SIZEOF(QUERY(g <* geometric_representation_context | NOT ((g.
        coordinate_space_dimension = 2) OR (g.coordinate_space_dimension = 3))
        )) = 0;
END_RULE;

RULE draughting_model_annotation_layers FOR (draughting_model);
 WHERE
  wr1 : SIZEOF(QUERY(dm <* draughting_model | NOT (SIZEOF(QUERY(ao <* QUERY(it
        <* dm\representation.items | (
        'AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE' IN TYPEOF(it))) | NOT (
        SIZEOF(USEDIN(ao,'AUTOMOTIVE_DESIGN.' +
        'PRESENTATION_LAYER_ASSIGNMENT.' + 'ASSIGNED_ITEMS')) >= 1))) = 0))) =
         0;
END_RULE;

RULE draughting_subfigure_representation_layers FOR (
     draughting_subfigure_representation);
 WHERE
  wr1 : SIZEOF(QUERY(dsr <* draughting_subfigure_representation | NOT (SIZEOF(
        QUERY(item <* dsr\representation.items | (
```

```
        'AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE' IN TYPEOF(item)) AND (SIZEOF
        (USEDIN(item,'AUTOMOTIVE_DESIGN.' + 'PRESENTATION_LAYER_ASSIGNMENT.' +
         'ASSIGNED_ITEMS')) = 0))) = 0))) = 0;
END_RULE;


RULE drawing_sheet_annotation_layers FOR (drawing_sheet_revision);
 WHERE
  wr1 : SIZEOF(QUERY(dsr <* drawing_sheet_revision | NOT (SIZEOF(QUERY(ao <*
        QUERY(it <* dsr\representation.items | (
        'AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE' IN TYPEOF(it))) | NOT (
        SIZEOF(USEDIN(ao,'AUTOMOTIVE_DESIGN.' +
        'PRESENTATION_LAYER_ASSIGNMENT.' + 'ASSIGNED_ITEMS')) >= 1))) = 0))) =
         0;
END_RULE;


RULE drawing_sheet_layout_usage FOR (mapped_item);
 WHERE
  wr1 : SIZEOF(QUERY(dsl <* QUERY(mi <* mapped_item | (
        'AUTOMOTIVE_DESIGN.DRAWING_SHEET_LAYOUT' IN TYPEOF(mi.mapping_source.
        mapped_representation))) | NOT (SIZEOF(USEDIN(dsl,'')) = SIZEOF(QUERY(
        dsr <* USEDIN(dsl,'AUTOMOTIVE_DESIGN.' + 'REPRESENTATION.ITEMS') | (
        'AUTOMOTIVE_DESIGN.DRAWING_SHEET_REVISION' IN TYPEOF(dsr))))))) = 0;
END_RULE;


RULE drawing_view_annotation_layers FOR (presentation_view);
 WHERE
  wr1 : SIZEOF(QUERY(pv <* presentation_view | NOT (SIZEOF(QUERY(ao <* QUERY(
        it <* pv\representation.items | (
        'AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE' IN TYPEOF(it))) | NOT (
        SIZEOF(USEDIN(ao,'AUTOMOTIVE_DESIGN.' +
        'PRESENTATION_LAYER_ASSIGNMENT.' + 'ASSIGNED_ITEMS')) >= 1))) = 0))) =
         0;
END_RULE;


RULE effective_context_assignment_subtype_exclusiveness FOR (
     effective_context_assignment);
 WHERE
  wr1 : SIZEOF(QUERY(eca <* effective_context_assignment | (SIZEOF(TYPEOF(eca)
         * ['AUTOMOTIVE_DESIGN.APPLIED_EFFECTIVE_CONTEXT_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.CONFIGURED_EFFECTIVE_CONTEXT_ASSIGNMENT']) > 1) OR
        ((SIZEOF(TYPEOF(eca)) > 2) AND (SIZEOF(TYPEOF(eca) * [
        'AUTOMOTIVE_DESIGN.APPLIED_EFFECTIVE_CONTEXT_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.CONFIGURED_EFFECTIVE_CONTEXT_ASSIGNMENT']) = 1))))
        = 0;
END_RULE;


RULE effective_item_assignment_subtype_exclusiveness FOR (
     effective_item_assignment);
 WHERE
```

```
  wr1 : SIZEOF(QUERY(eia <* effective_item_assignment | (SIZEOF(TYPEOF(eia) *
        ['AUTOMOTIVE_DESIGN.APPLIED_EFFECTICE_ITEM_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.CONFIGURED_EFFECTIVE_ITEM_ASSIGNMENT']) > 1) OR ((
        SIZEOF(TYPEOF(eia)) > 2) AND (SIZEOF(TYPEOF(eia) * [
        'AUTOMOTIVE_DESIGN.APPLIED_EFFECTICE_ITEM_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.CONFIGURED_EFFECTIVE_ITEM_ASSIGNMENT']) = 1)))) = 0
        ;
END_RULE;


RULE effectivity_requires_effective_item_assignment FOR (effectivity);
 WHERE
  wr1 : SIZEOF(QUERY(e <* effectivity | SIZEOF(USEDIN(e,'AUTOMOTIVE_DESIGN.' +
         'EFFECTIVE_ITEM_ASSIGNMENT.ASSIGNED_EFFECTIVITY')) < 1)) = 0;
END_RULE;


RULE fill_area_style_tile_symbol_constraint FOR (
     fill_area_style_tile_symbol_with_style);
 WHERE
  wr1 : SIZEOF(QUERY(fast <* fill_area_style_tile_symbol_with_style | ((
        'AUTOMOTIVE_DESIGN.DEFINED_SYMBOL' IN TYPEOF(fast.symbol\styled_item.
        item)) AND NOT ('AUTOMOTIVE_DESIGN.EXTERNALLY_DEFINED_SYMBOL' IN
        TYPEOF(fast.symbol\styled_item.item\defined_symbol.definition))))) = 0
        ;
END_RULE;


RULE functional_product_definition_requires_functional_product FOR (
     product_definition);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* product_definition | (pd.frame_of_reference.name =
        'functional definition') AND (SIZEOF(QUERY(pc <* pd.formation.
        of_product.frame_of_reference | pc.name = 'functionality')) < 1))) = 0
        ;
END_RULE;


RULE geometric_tolerance_subtype_exclusiveness FOR (geometric_tolerance);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* geometric_tolerance | (SIZEOF(TYPEOF(pd) * [
        'AUTOMOTIVE_DESIGN.ANGULARITY_TOLERANCE',
        'AUTOMOTIVE_DESIGN.CIRCULAR_RUNOUT_TOLERANCE',
        'AUTOMOTIVE_DESIGN.CIRCULARITY_TOLERANCE',
        'AUTOMOTIVE_DESIGN.CONCENTRICITY_TOLERANCE',
        'AUTOMOTIVE_DESIGN.CYLINDRICITY_TOLERANCE',
        'AUTOMOTIVE_DESIGN.FLATNESS_TOLERANCE',
        'AUTOMOTIVE_DESIGN.LINEAR_PROFILE_TOLERANCE',
        'AUTOMOTIVE_DESIGN.PARALLELISM_TOLERANCE',
        'AUTOMOTIVE_DESIGN.PERPENDICULARITY_TOLERANCE',
        'AUTOMOTIVE_DESIGN.POSITION_TOLERANCE',
        'AUTOMOTIVE_DESIGN.STRAIGHTNESS_TOLERANCE',
        'AUTOMOTIVE_DESIGN.SURFACE_PROFILE_TOLERANCE',
        'AUTOMOTIVE_DESIGN.SYMMETRY_TOLERANCE',
```

```
        'AUTOMOTIVE_DESIGN.TOTAL_RUNOUT_TOLERANCE']) > 1) OR ((SIZEOF(TYPEOF(
        pd)) > 2) AND (SIZEOF(TYPEOF(pd) * [
        'AUTOMOTIVE_DESIGN.ANGULARITY_TOLERANCE',
        'AUTOMOTIVE_DESIGN.CIRCULAR_RUNOUT_TOLERANCE',
        'AUTOMOTIVE_DESIGN.CIRCULARITY_TOLERANCE',
        'AUTOMOTIVE_DESIGN.CONCENTRICITY_TOLERANCE',
        'AUTOMOTIVE_DESIGN.CYLINDRICITY_TOLERANCE',
        'AUTOMOTIVE_DESIGN.FLATNESS_TOLERANCE',
        'AUTOMOTIVE_DESIGN.LINEAR_PROFILE_TOLERANCE',
        'AUTOMOTIVE_DESIGN.PARALLELISM_TOLERANCE',
        'AUTOMOTIVE_DESIGN.PERPENDICULARITY_TOLERANCE',
        'AUTOMOTIVE_DESIGN.POSITION_TOLERANCE',
        'AUTOMOTIVE_DESIGN.STRAIGHTNESS_TOLERANCE',
        'AUTOMOTIVE_DESIGN.SURFACE_PROFILE_TOLERANCE',
        'AUTOMOTIVE_DESIGN.SYMMETRY_TOLERANCE',
        'AUTOMOTIVE_DESIGN.TOTAL_RUNOUT_TOLERANCE']) = 1)))) = 0;
END_RULE;


RULE global_length_and_angle_units_2d_or_3d FOR (global_unit_assigned_context)
                                                 ;
 WHERE
  wr1 : SIZEOF(QUERY(guac <* global_unit_assigned_context | NOT (SIZEOF(guac.
        units) >= 2))) = 0;
  wr2 : SIZEOF(QUERY(guac <* global_unit_assigned_context | NOT ((SIZEOF(QUERY
        (x <* guac.units | 'AUTOMOTIVE_DESIGN.LENGTH_UNIT' IN TYPEOF(x))) = 1)
         AND (SIZEOF(QUERY(x <* guac.units |
        'AUTOMOTIVE_DESIGN.PLANE_ANGLE_UNIT' IN TYPEOF(x))) = 1)))) = 0;
END_RULE;


RULE group_assignment_subtype_exclusiveness FOR (group_assignment);
 WHERE
  wr1 : SIZEOF(QUERY(ga <* group_assignment | (SIZEOF(TYPEOF(ga) * [
        'AUTOMOTIVE_DESIGN.APPLIED_GROUP_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.GROUP_MEMBER_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_FEATURE_CATEGORY_USAGE']) > 1) OR (
        (SIZEOF(TYPEOF(ga)) > 2) AND (SIZEOF(TYPEOF(ga) * [
        'AUTOMOTIVE_DESIGN.APPLIED_GROUP_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.GROUP_MEMBER_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.PRODUCT_CONCEPT_FEATURE_CATEGORY_USAGE']) = 1)))) =
         0;
END_RULE;


RULE identification_assignment_subtype_exclusiveness FOR (
     identification_assignment);
 WHERE
  wr1 : SIZEOF(QUERY(ia <* identification_assignment | (SIZEOF(TYPEOF(ia) * [
        'AUTOMOTIVE_DESIGN.APPLIED_IDENTIFICATION_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT']) > 1) OR ((SIZEOF(TYPEOF(ia))
        > 2) AND (SIZEOF(TYPEOF(ia) * [
```

```
        'AUTOMOTIVE_DESIGN.APPLIED_IDENTIFICATION_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT']) = 1)))) = 0;
END_RULE;


RULE organization_assignment_subtype_exclusiveness FOR (
     organization_assignment);
 WHERE
  wr1 : SIZEOF(QUERY(oa <* organization_assignment | (SIZEOF(TYPEOF(oa) * [
        'AUTOMOTIVE_DESIGN.APPLIED_DATE_AND_ORGANIZATION_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.APPLIED_ORGANIZATION_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.PROJECT_ASSIGNMENT']) > 1) OR ((SIZEOF(TYPEOF(oa))
        > 2) AND (SIZEOF(TYPEOF(oa) * [
        'AUTOMOTIVE_DESIGN.APPLIED_DATE_AND_ORGANIZATION_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.APPLIED_ORGANIZATION_ASSIGNMENT',
        'AUTOMOTIVE_DESIGN.PROJECT_ASSIGNMENT']) = 1)))) = 0;
END_RULE;


RULE person_requires_person_and_organization FOR (person);
 WHERE
  wr1 : SIZEOF(QUERY(p <* person | SIZEOF(USEDIN(p,'AUTOMOTIVE_DESIGN.' +
        'PERSON_AND_ORGANIZATION.THE_PERSON')) < 1)) = 0;
END_RULE;


RULE presentation_layer_assignment_constraint_2d_or_3d FOR (
     presentation_layer_assignment);
 WHERE
  wr1 : SIZEOF(QUERY(pla <* presentation_layer_assignment | NOT (SIZEOF(QUERY(
        item <* pla.assigned_items | NOT (SIZEOF(TYPEOF(item) * [
        'AUTOMOTIVE_DESIGN.POINT','AUTOMOTIVE_DESIGN.CURVE',
        'AUTOMOTIVE_DESIGN.SURFACE','AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE']
        ) = 1))) = 0))) = 0;
  wr2 : SIZEOF(QUERY(pla <* presentation_layer_assignment | NOT (SIZEOF(QUERY(
        pnt <* QUERY(item <* pla.assigned_items | ('AUTOMOTIVE_DESIGN.POINT'
        IN TYPEOF(item))) | NOT (SIZEOF(QUERY(rep <* USEDIN(pnt,
        'AUTOMOTIVE_DESIGN.REPRESENTATION.ITEMS') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN TYPEOF(rep)))) = 0))) = 0)
        )) = 0;
  wr3 : SIZEOF(QUERY(pla <* presentation_layer_assignment | NOT (SIZEOF(QUERY(
        crv <* QUERY(item <* pla.assigned_items | ('AUTOMOTIVE_DESIGN.CURVE'
        IN TYPEOF(item))) | NOT (SIZEOF(QUERY(rep <* USEDIN(crv,
        'AUTOMOTIVE_DESIGN.REPRESENTATION.ITEMS') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN TYPEOF(rep)))) = 0))) = 0)
        )) = 0;
  wr4 : SIZEOF(QUERY(pla <* presentation_layer_assignment | NOT (SIZEOF(QUERY(
        srf <* QUERY(item <* pla.assigned_items | ('AUTOMOTIVE_DESIGN.SURFACE'
         IN TYPEOF(item))) | NOT (SIZEOF(QUERY(rep <* USEDIN(srf,
        'AUTOMOTIVE_DESIGN.REPRESENTATION.ITEMS') | (
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION' IN TYPEOF(rep)))) = 0))) = 0)
        )) = 0;
```

```
  wr5 : SIZEOF(QUERY(pla <* presentation_layer_assignment | NOT (SIZEOF(QUERY(
        ao <* QUERY(item <* pla.assigned_items | (
        'AUTOMOTIVE_DESIGN.ANNOTATION_OCCURRENCE' IN TYPEOF(item))) | NOT (
        SIZEOF(QUERY(ur <* using_representations(ao) | NOT (SIZEOF(TYPEOF(ur)
        * ['AUTOMOTIVE_DESIGN.' + 'DRAUGHTING_SUBFIGURE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.DRAUGHTING_MODEL',
        'AUTOMOTIVE_DESIGN.PRESENTATION_AREA',
        'AUTOMOTIVE_DESIGN.PRESENTATION_VIEW']) = 1))) = 0))) = 0))) = 0;
END_RULE;


RULE product_definition_substitute_requires_effectivity_assignment FOR (
     applied_effective_item_assignment,product_definition_substitute);
 WHERE
  wr1 : SIZEOF(QUERY(pds <* product_definition_substitute | SIZEOF(QUERY(aeia
        <* USEDIN(pds,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_EFFECTIVE_ITEM_ASSIGNMENT.ITEMS') | (
        'AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY' IN TYPEOF(aeia.
        assigned_effectivity)) OR ((
        'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY' IN TYPEOF(aeia.
        assigned_effectivity)) AND (SIZEOF(QUERY(ea <* USEDIN(aeia.
        assigned_effectivity,'AUTOMOTIVE_DESIGN.EVENT_ASSIGNMENT.ITEMS') | ea.
        role.name = 'effectivity start')) = 1)))) = 0)) = 0;
END_RULE;


RULE product_definition_subtype_exclusiveness FOR (product_definition);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* product_definition | (SIZEOF(TYPEOF(pd) * [
        'AUTOMOTIVE_DESIGN.MATED_ITEM_RELATIONSHIP',
        'AUTOMOTIVE_DESIGN.PHYSICALLY_MODELLED_PRODUCT_DEFINITION',
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RESOURCE',
        'AUTOMOTIVE_DESIGN.SOLUTION_BASED_PRODUCT_DEFINITION']) > 1) OR ((
        SIZEOF(TYPEOF(pd)) > 2) AND (SIZEOF(TYPEOF(pd) * [
        'AUTOMOTIVE_DESIGN.MATED_ITEM_RELATIONSHIP',
        'AUTOMOTIVE_DESIGN.PHYSICALLY_MODELLED_PRODUCT_DEFINITION',
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RESOURCE',
        'AUTOMOTIVE_DESIGN.SOLUTION_BASED_PRODUCT_DEFINITION']) = 1)))) = 0;
END_RULE;


RULE product_requires_version FOR (product);
 WHERE
  wr1 : SIZEOF(QUERY(p <* product | SIZEOF(USEDIN(p,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_DEFINITION_FORMATION.OF_PRODUCT')) < 1)) = 0;
END_RULE;


RULE representation_subtype_exclusiveness FOR (representation);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* representation | (SIZEOF(TYPEOF(pd) * [
        'AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION',
```

```
        'AUTOMOTIVE_DESIGN.CONSTRUCTIVE_GEOMETRY_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.DEFAULT_TOLERANCE_TABLE',
        'AUTOMOTIVE_DESIGN.DEFAULT_TOLERANCE_TABLE_CELL',
        'AUTOMOTIVE_DESIGN.DEFINITIONAL_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.DRAUGHTING_MODEL',
        'AUTOMOTIVE_DESIGN.EXTERNALLY_DEFINED_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.KINEMATIC_FRAME_BACKGROUND_REPRESENTATION ',
        'AUTOMOTIVE_DESIGN.KINEMATIC_GROUND_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.KINEMATIC_LINK_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.LOCATION_SHAPE_REPRESENTATION',
  'AUTOMOTIVE_DESIGN.MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.PATH_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.PRESENTATION_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.PROPERTY_VALUE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.SYMBOL_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.TEXT_STRING_REPRESENTATION']) > 1) OR ((SIZEOF(
        TYPEOF(pd)) > 2) AND (SIZEOF(TYPEOF(pd) * [
        'AUTOMOTIVE_DESIGN.CONNECTED_FACE_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.CONSTRUCTIVE_GEOMETRY_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.DEFAULT_TOLERANCE_TABLE',
        'AUTOMOTIVE_DESIGN.DEFAULT_TOLERANCE_TABLE_CELL',
        'AUTOMOTIVE_DESIGN.DEFINITIONAL_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.DIRECTION_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.DRAUGHTING_MODEL',
        'AUTOMOTIVE_DESIGN.EXTERNALLY_DEFINED_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.FACE_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.KINEMATIC_FRAME_BACKGROUND_REPRESENTATION ',
        'AUTOMOTIVE_DESIGN.KINEMATIC_GROUND_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.KINEMATIC_LINK_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.LOCATION_SHAPE_REPRESENTATION',
  'AUTOMOTIVE_DESIGN.MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.PATH_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.PRESENTATION_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.PROPERTY_VALUE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.SYMBOL_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.TEXT_STRING_REPRESENTATION']) = 1)))) = 0;
END_RULE;


RULE restrict_action_directive FOR (action_directive);
 WHERE
  wr1 : SIZEOF(QUERY(ad <* action_directive | SIZEOF(USEDIN(ad,
        'AUTOMOTIVE_DESIGN.DIRECTED_ACTION.DIRECTIVE')) < 1)) = 0;
  wr2 : SIZEOF(QUERY(ad <* action_directive | (ad.analysis <> 'internal') AND
        (ad.analysis <> 'external'))) = 0;
END_RULE;


RULE restrict_action_resource_requirement_for_process_operation FOR (action,
```

```
     action_resource_requirement);
 WHERE
  wr1 : SIZEOF(QUERY(a <* action | (a.description = 'process operation') AND (
        SIZEOF(QUERY(arr <* USEDIN(a,'AUTOMOTIVE_DESIGN.' +
        'ACTION_RESOURCE_REQUIREMENT.OPERATIONS') | arr.name =
        'reference tool')) > 1))) = 0;
  wr2 : SIZEOF(QUERY(arr <* action_resource_requirement | (arr.name =
        'reference tool') AND (NOT (
        'AUTOMOTIVE_DESIGN.REQUIREMENT_FOR_ACTION_RESOURCE' IN TYPEOF(arr)))
        AND (SIZEOF(QUERY(ar <* arr\requirement_for_action_resource.resources
         | 'AUTOMOTIVE_DESIGN.PRODUCT_DEFINITION_RESOURCE' IN TYPEOF(ar))) <>
        1))) = 0;
  wr3 : SIZEOF(QUERY(arr <* action_resource_requirement | (arr.name <>
        'reference tool') AND (arr.name <> 'non reference tool'))) = 0;
END_RULE;


RULE restrict_approval_date_time_for_approval FOR (approval);
 WHERE
  wr1 : SIZEOF(QUERY(a <* approval | SIZEOF(QUERY(adt <* USEDIN(a,
        'AUTOMOTIVE_DESIGN.' + 'APPROVAL_DATE_TIME.DATED_APPROVAL') | adt.role
        .name = 'actual')) > 1)) = 0;
  wr2 : SIZEOF(QUERY(a <* approval | SIZEOF(QUERY(adt <* USEDIN(a,
        'AUTOMOTIVE_DESIGN.' + 'APPROVAL_DATE_TIME.DATED_APPROVAL') | adt.role
        .name = 'nominal')) > 1)) = 0;
END_RULE;


RULE restrict_approval_date_time_role FOR (approval_date_time_role);
 WHERE
  wr1 : SIZEOF(QUERY(adtr <* approval_date_time_role | (adtr.name <> 'actual')
         AND (adtr.name <> 'nominal'))) = 0;
END_RULE;


RULE restrict_classification_system_for_approval_status FOR (approval_status,
     group_member_assignment);
 WHERE
  wr1 : SIZEOF(QUERY(a_s <* approval_status | SIZEOF(USEDIN(a_s,
        'AUTOMOTIVE_DESIGN.' + 'GROUP_MEMBER_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_classification_system_for_document_type FOR (document_type,
     group_member_assignment);
 WHERE
  wr1 : SIZEOF(QUERY(dt <* document_type | SIZEOF(USEDIN(dt,
        'AUTOMOTIVE_DESIGN.' + 'GROUP_MEMBER_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_classification_system_for_security_classification_level FOR (
     group_member_assignment,security_classification_level);
 WHERE
```

```
  wr1 : SIZEOF(QUERY(scl <* security_classification_level | SIZEOF(USEDIN(scl,
        'AUTOMOTIVE_DESIGN.' + 'GROUP_MEMBER_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_multi_language_for_action FOR (action);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* action | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'description') AND (amlaa.attribute_name <> 'name'))
        ) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* action | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_action_directive FOR (action_directive);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* action_directive | SIZEOF(QUERY(amlaa <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'comment')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* action_directive | SIZEOF(QUERY(amlaa1 <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_action_method FOR (action_method);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* action_method | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'description') AND (amlaa.attribute_name <> 'name')
        AND (amlaa.attribute_name <> 'purpose') AND (amlaa.attribute_name <>
        'consequence'))) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* action_method | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
```

```
END_RULE;

RULE restrict_multi_language_for_action_property FOR (action_property);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* action_property | SIZEOF(QUERY(amlaa <* USEDIN(ent
        ,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* action_property | SIZEOF(QUERY(amlaa1 <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;

RULE restrict_multi_language_for_action_relationship FOR (action_relationship)
                                                          ;
 WHERE
  wr1 : SIZEOF(QUERY(ent <* action_relationship | SIZEOF(QUERY(amlaa <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* action_relationship | SIZEOF(QUERY(amlaa1 <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;

RULE restrict_multi_language_for_action_resource_requirement FOR (
     action_resource_requirement);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* action_resource_requirement | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* action_resource_requirement | SIZEOF(QUERY(amlaa1
        <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;

RULE restrict_multi_language_for_alternate_product_relationship FOR (
```

```
     alternate_product_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* alternate_product_relationship | SIZEOF(QUERY(
        amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'basis')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* alternate_product_relationship | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_application_context FOR (application_context)
                                                          ;
 WHERE
  wr1 : SIZEOF(QUERY(ent <* application_context | SIZEOF(QUERY(amlaa <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* application_context | SIZEOF(QUERY(amlaa1 <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_approval FOR (approval);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* approval | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'level')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* approval | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_assembly_component_usage FOR (
     assembly_component_usage);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* assembly_component_usage | SIZEOF(QUERY(amlaa <*
```

```
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'product_definition_relationship.description')) > 0)
        ) = 0;
  wr2 : SIZEOF(QUERY(ent <* assembly_component_usage | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_assembly_component_usage_substitute FOR (
     assembly_component_usage_substitute);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* assembly_component_usage_substitute | SIZEOF(QUERY
        (amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* assembly_component_usage_substitute | SIZEOF(QUERY
        (amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_camera_image_2d_with_scale FOR (
     camera_image_2d_with_scale);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* camera_image_2d_with_scale | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'representation_item.name')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* camera_image_2d_with_scale | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_camera_image_3d_with_scale FOR (
     camera_image_3d_with_scale);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* camera_image_3d_with_scale | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
```

```
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'representation_item.name')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* camera_image_3d_with_scale | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_certification FOR (certification);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* certification | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'name') AND (amlaa.attribute_name <> 'purpose'))) >
        0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* certification | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_characterized_product_category FOR (
     characterized_product_category);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* characterized_product_category | SIZEOF(QUERY(
        amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'product_category.name')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* characterized_product_category | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_classification_system FOR (
     classification_system);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* classification_system | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'group.description') AND (amlaa.attribute_name <>
```

```
        'group.name'))) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* classification_system | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_configurable_item FOR (configurable_item);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* configurable_item | SIZEOF(QUERY(amlaa <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'configuration_item.name')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* configurable_item | SIZEOF(QUERY(amlaa1 <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_configuration_design FOR (
     configuration_design);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* configuration_design | SIZEOF(QUERY(amlaa <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* configuration_design | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_configuration_item_relationship FOR (
     configuration_item_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* configuration_item_relationship | SIZEOF(QUERY(
        amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* configuration_item_relationship | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
```

```
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE
 restrict_multi_language_for_constructive_geometry_representation_relationship
      FOR (constructive_geometry_representation_relationship);
  WHERE
   wr1 : SIZEOF(QUERY(ent <* constructive_geometry_representation_relationship
          | SIZEOF(QUERY(amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'representation_relationship.description')) > 0)) =
        0;
   wr2 : SIZEOF(QUERY(ent <* constructive_geometry_representation_relationship
          | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_data_environment FOR (data_environment);
  WHERE
   wr1 : SIZEOF(QUERY(ent <* data_environment | SIZEOF(QUERY(amlaa <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
   wr2 : SIZEOF(QUERY(ent <* data_environment | SIZEOF(QUERY(amlaa1 <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_default_tolerance_table_cell FOR (
      default_tolerance_table_cell);
  WHERE
   wr1 : SIZEOF(QUERY(ent <* default_tolerance_table_cell | SIZEOF(QUERY(amlaa
        <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'representation.description')) > 0)) = 0;
   wr2 : SIZEOF(QUERY(ent <* default_tolerance_table_cell | SIZEOF(QUERY(amlaa1
        <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
```

```
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_descriptive_representation_item FOR (
     descriptive_representation_item);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* descriptive_representation_item | SIZEOF(QUERY(
        amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* descriptive_representation_item | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_dimensional_location FOR (
     dimensional_location);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* dimensional_location | SIZEOF(QUERY(amlaa <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'shape_aspect_relationship.description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* dimensional_location | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_dimensional_size FOR (dimensional_size);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* dimensional_size | SIZEOF(QUERY(amlaa <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'applies_to.shape_aspect.description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* dimensional_size | SIZEOF(QUERY(amlaa1 <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
```

```
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_document FOR (document);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* document | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'description') AND (amlaa.attribute_name <> 'name'))
        ) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* document | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_document_relationship FOR (
     document_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* document_relationship | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* document_relationship | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_draughting_model FOR (draughting_model);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* draughting_model | SIZEOF(QUERY(amlaa <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'representation.description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* draughting_model | SIZEOF(QUERY(amlaa1 <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;
```

```
RULE restrict_multi_language_for_draughting_title FOR (draughting_title);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* draughting_title | SIZEOF(QUERY(amlaa <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'contents')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* draughting_title | SIZEOF(QUERY(amlaa1 <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_drawing_sheet_revision FOR (
     drawing_sheet_revision);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* drawing_sheet_revision | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'representation.name')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* drawing_sheet_revision | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_effectivity FOR (effectivity);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* effectivity | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* effectivity | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_effectivity_relationship FOR (
     effectivity_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* effectivity_relationship | SIZEOF(QUERY(amlaa <*
```

```
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* effectivity_relationship | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_externally_defined_classification_attribute
     FOR (externally_defined_classification_attribute);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* externally_defined_classification_attribute |
        SIZEOF(QUERY(amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'representation.name')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* externally_defined_classification_attribute |
        SIZEOF(QUERY(amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_externally_defined_classification_system FOR (
     externally_defined_classification_system);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* externally_defined_classification_system | SIZEOF(
        QUERY(amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'group.description') AND (amlaa.attribute_name <>
        'group.name'))) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* externally_defined_classification_system | SIZEOF(
        QUERY(amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_externally_defined_product_category FOR (
     externally_defined_product_category);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* externally_defined_product_category | SIZEOF(QUERY
        (amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
```

```
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'product_category.name')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* externally_defined_product_category | SIZEOF(QUERY
        (amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_externally_defined_representation FOR (
     externally_defined_representation);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* externally_defined_representation | SIZEOF(QUERY(
        amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* externally_defined_representation | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_geometric_tolerance FOR (geometric_tolerance)
                                                          ;
 WHERE
  wr1 : SIZEOF(QUERY(ent <* geometric_tolerance | SIZEOF(QUERY(amlaa <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'name') AND (amlaa.attribute_name <> 'description'))
        ) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* geometric_tolerance | SIZEOF(QUERY(amlaa1 <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_group FOR (group);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* group | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'name') AND (amlaa.attribute_name <> 'description'))
```

```
        ) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* group | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_identification_role FOR (identification_role)
                                                                  ;
 WHERE
  wr1 : SIZEOF(QUERY(ent <* identification_role | SIZEOF(QUERY(amlaa <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* identification_role | SIZEOF(QUERY(amlaa1 <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_make_from_usage_option FOR (
     make_from_usage_option);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* make_from_usage_option | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'product_definition_relationship.description') AND (
        amlaa.attribute_name <> 'product_definition_relationship.name'))) > 0)
        ) = 0;
  wr2 : SIZEOF(QUERY(ent <* make_from_usage_option | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_material_property FOR (material_property);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* material_property | SIZEOF(QUERY(amlaa <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'property_definition.description')) > 0)) = 0;
```

```
   wr2 : SIZEOF(QUERY(ent <* material_property | SIZEOF(QUERY(amlaa1 <* USEDIN(
         ent,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
         amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
         attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
         language))) > 1)) > 0)) = 0;
 END_RULE;


 RULE restrict_multi_language_for_measure_representation_item FOR (
      measure_representation_item);
  WHERE
   wr1 : SIZEOF(QUERY(ent <* measure_representation_item | SIZEOF(QUERY(amlaa <*
         USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
         attribute_name <> 'representation_item.name')) > 0)) = 0;
   wr2 : SIZEOF(QUERY(ent <* measure_representation_item | SIZEOF(QUERY(amlaa1
         <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
         amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
         attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
         language))) > 1)) > 0)) = 0;
 END_RULE;


RULE
 restrict_multi_language_for_mechanical_design_geometric_presentation_representation
      FOR (mechanical_design_geometric_presentation_representation);
  WHERE
   wr1 : SIZEOF(QUERY(ent <*
         mechanical_design_geometric_presentation_representation | SIZEOF(QUERY
         (amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
         attribute_name <> 'representation.description')) > 0)) = 0;
   wr2 : SIZEOF(QUERY(ent <*
         mechanical_design_geometric_presentation_representation | SIZEOF(QUERY
         (amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
         amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
         attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
         language))) > 1)) > 0)) = 0;
 END_RULE;


 RULE restrict_multi_language_for_model_change FOR (model_change);
  WHERE
   wr1 : SIZEOF(QUERY(ent <* model_change | SIZEOF(QUERY(amlaa <* USEDIN(ent,
         'AUTOMOTIVE_DESIGN.' +
         'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
         attribute_name <> 'property_definition.description')) > 0)) = 0;
```

```
  wr2 : SIZEOF(QUERY(ent <* model_change | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_organization_relationship FOR (
     organization_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* organization_relationship | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* organization_relationship | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_organization_role FOR (organization_role);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* organization_role | SIZEOF(QUERY(amlaa <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'name')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* organization_role | SIZEOF(QUERY(amlaa1 <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_pair_actuator FOR (pair_actuator);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* pair_actuator | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'name')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* pair_actuator | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
```

```
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_presentation_area FOR (presentation_area);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* presentation_area | SIZEOF(QUERY(amlaa <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'representation.description') AND (amlaa.
        attribute_name <> 'representation.name'))) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* presentation_area | SIZEOF(QUERY(amlaa1 <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_presentation_layer_assignment FOR (
     presentation_layer_assignment);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* presentation_layer_assignment | SIZEOF(QUERY(amlaa
        <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* presentation_layer_assignment | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_presentation_view FOR (presentation_view);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* presentation_view | SIZEOF(QUERY(amlaa <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'representation.description') AND (amlaa.
        attribute_name <> 'representation.name'))) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* presentation_view | SIZEOF(QUERY(amlaa1 <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
```

```
        language))) > 1)) > 0)) = 0;
END_RULE;

RULE restrict_multi_language_for_presentation_view_relationship FOR (
     presentation_view_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* presentation_view_relationship | SIZEOF(QUERY(
        amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'representation_relationship.description')) > 0)) =
        0;
  wr2 : SIZEOF(QUERY(ent <* presentation_view_relationship | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;

RULE restrict_multi_language_for_process_change FOR (process_change);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* process_change | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'action_property.description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* process_change | SIZEOF(QUERY(amlaa1 <* USEDIN(ent
        ,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;

RULE restrict_multi_language_for_process_plan FOR (process_plan);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* process_plan | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'action.description') AND (amlaa.attribute_name <>
        'action.name'))) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* process_plan | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;
```

```
RULE restrict_multi_language_for_process_product_association FOR (
     process_product_association);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* process_product_association | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* process_product_association | SIZEOF(QUERY(amlaa1
        <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product FOR (product);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'description') AND (amlaa.attribute_name <> 'name'))
        ) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* product | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product_category FOR (product_category);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_category | SIZEOF(QUERY(amlaa <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* product_category | SIZEOF(QUERY(amlaa1 <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product_class FOR (product_class);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_class | SIZEOF(QUERY(amlaa <* USEDIN(ent,
```

```
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'product_concept.description') AND (amlaa.
        attribute_name <> 'product_concept.name'))) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* product_class | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product_component FOR (product_component);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_component | SIZEOF(QUERY(amlaa <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'configuration_item.description') AND (amlaa.
        attribute_name <> 'configuration_item.name'))) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* product_component | SIZEOF(QUERY(amlaa1 <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product_concept_feature FOR (
     product_concept_feature);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_concept_feature | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'description') AND (amlaa.attribute_name <> 'name'))
        ) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* product_concept_feature | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product_concept_feature_association FOR (
     product_concept_feature_association);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_concept_feature_association | SIZEOF(QUERY
```

```
        (amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* product_concept_feature_association | SIZEOF(QUERY
        (amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product_concept_feature_category FOR (
     product_concept_feature_category);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_concept_feature_category | SIZEOF(QUERY(
        amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'group.description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* product_concept_feature_category | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product_definition FOR (product_definition);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_definition | SIZEOF(QUERY(amlaa <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'name') AND (amlaa.attribute_name <> 'description'))
        ) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* product_definition | SIZEOF(QUERY(amlaa1 <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product_definition_formation FOR (
     product_definition_formation);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_definition_formation | SIZEOF(QUERY(amlaa
        <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
```

```
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* product_definition_formation | SIZEOF(QUERY(amlaa1
        <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product_definition_formation_relationship FOR
     (product_definition_formation_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_definition_formation_relationship | SIZEOF
        (QUERY(amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* product_definition_formation_relationship | SIZEOF
        (QUERY(amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product_definition_occurrence FOR (
     product_definition_occurrence);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_definition_occurrence | SIZEOF(QUERY(amlaa
        <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'product_definition.description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* product_definition_occurrence | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product_definition_relationship FOR (
     product_definition_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_definition_relationship | SIZEOF(QUERY(
        amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'description') AND (amlaa.attribute_name <> 'id')))
        > 0)) = 0;
```

```
  wr2 : SIZEOF(QUERY(ent <* product_definition_relationship | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product_definition_shape FOR (
     product_definition_shape);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_definition_shape | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'property_definition.description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* product_definition_shape | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_product_definition_substitute FOR (
     product_definition_substitute);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_definition_substitute | SIZEOF(QUERY(amlaa
        <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* product_definition_substitute | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_project FOR (project);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* project | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'organization.description') AND (amlaa.
        attribute_name <> 'organization.name'))) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* project | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
```

```
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_property_change FOR (property_change);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* property_change | SIZEOF(QUERY(amlaa <* USEDIN(ent
        ,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'property_definition.description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* property_change | SIZEOF(QUERY(amlaa1 <* USEDIN(
        ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_property_definition FOR (property_definition)
                                                          ;
 WHERE
  wr1 : SIZEOF(QUERY(ent <* property_definition | SIZEOF(QUERY(amlaa <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* property_definition | SIZEOF(QUERY(amlaa1 <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_property_definition_relationship FOR (
     property_definition_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* property_definition_relationship | SIZEOF(QUERY(
        amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* property_definition_relationship | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
```

```
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_property_definition_representation FOR (
     property_definition_representation);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* property_definition_representation | SIZEOF(QUERY(
        amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'name')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* property_definition_representation | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_rep_item_group FOR (rep_item_group);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* rep_item_group | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'group.name') AND (amlaa.attribute_name <>
        'group.description'))) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* rep_item_group | SIZEOF(QUERY(amlaa1 <* USEDIN(ent
        ,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_representation FOR (representation);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* representation | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa.
        attribute_name <> 'name') AND (amlaa.attribute_name <> 'description'))
        ) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* representation | SIZEOF(QUERY(amlaa1 <* USEDIN(ent
        ,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
```

```
END_RULE;

RULE restrict_multi_language_for_representation_item FOR (representation_item)
                                                                  ;
 WHERE
  wr1 : SIZEOF(QUERY(ent <* representation_item | SIZEOF(QUERY(amlaa <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'name')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* representation_item | SIZEOF(QUERY(amlaa1 <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;

RULE restrict_multi_language_for_representation_relationship FOR (
     representation_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* representation_relationship | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* representation_relationship | SIZEOF(QUERY(amlaa1
        <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;

RULE restrict_multi_language_for_resource_requirement_type FOR (
     resource_requirement_type);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* resource_requirement_type | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'name')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* resource_requirement_type | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;
```

```
RULE restrict_multi_language_for_retention FOR (retention);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* retention | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'action.description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* retention | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_security_classification FOR (
     security_classification);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* security_classification | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'purpose')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* security_classification | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_shape_aspect FOR (shape_aspect);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* shape_aspect | SIZEOF(QUERY(amlaa <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* shape_aspect | SIZEOF(QUERY(amlaa1 <* USEDIN(ent,
        'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_shape_aspect_relationship FOR (
     shape_aspect_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* shape_aspect_relationship | SIZEOF(QUERY(amlaa <*
```

```
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* shape_aspect_relationship | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_shape_dimension_representation FOR (
     shape_dimension_representation);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* shape_dimension_representation | SIZEOF(QUERY(
        amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'representation.description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* shape_dimension_representation | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_shape_representation FOR (
     shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* shape_representation | SIZEOF(QUERY(amlaa <* USEDIN
        (ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'representation.description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* shape_representation | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_solution_based_product_definition FOR (
     solution_based_product_definition);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* solution_based_product_definition | SIZEOF(QUERY(
        amlaa <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
```

```
        attribute_name <> 'product_definition.description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* solution_based_product_definition | SIZEOF(QUERY(
        amlaa1 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_multi_language_for_versioned_action_request FOR (
     versioned_action_request);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* versioned_action_request | SIZEOF(QUERY(amlaa <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | amlaa.
        attribute_name <> 'description')) > 0)) = 0;
  wr2 : SIZEOF(QUERY(ent <* versioned_action_request | SIZEOF(QUERY(amlaa1 <*
        USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | SIZEOF(QUERY(
        amlaa2 <* USEDIN(ent,'AUTOMOTIVE_DESIGN.' +
        'APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT.ITEMS') | (amlaa1.
        attribute_name = amlaa2.attribute_name) AND (amlaa1.language = amlaa2.
        language))) > 1)) > 0)) = 0;
END_RULE;


RULE restrict_process_product_association FOR (process_product_association);
 WHERE
  wr1 : SIZEOF(QUERY(ppa <* process_product_association | (ppa.name <> 'input'
        ) AND (ppa.name <> 'output') AND (ppa.name <> 'produced output'))) = 0
        ;
  wr2 : SIZEOF(QUERY(ppa <* process_product_association | ((ppa.name = 'input'
        ) OR (ppa.name = 'output')) AND (SIZEOF(QUERY(ppai <* USEDIN(ppa,
        'AUTOMOTIVE_DESIGN.' + 'PROCESS_PLAN_ASSIGNMENT.ITEMS') | (ppai.name =
         'process plan context') AND ('AUTOMOTIVE_DESIGN.PROCESS_PLAN' IN
        TYPEOF(ppai.assigned_action)) AND (SIZEOF(QUERY(ar <* USEDIN(ppai.
        assigned_action,
        'AUTOMOTIVE_DESIGN.ACTION_RELATIONSHIP.RELATING_ACTION') | (ar.name =
        'process plan operation assignment') AND (ar.related_action :=: ppa.
        process) AND (ar.related_action.name = 'process operation'))) = 1)))
        <> 1))) = 0;
  wr3 : SIZEOF(QUERY(ppa <* process_product_association | (ppa.name =
        'produced output') AND (NOT ('AUTOMOTIVE_DESIGN.PROCESS_PLAN' IN
        TYPEOF(ppa.process))))) = 0;
END_RULE;


RULE restrict_product_categories_for_tool_part_relationship FOR (
     product_definition_relationship,product_related_product_category);
 WHERE
  wr1 : SIZEOF(QUERY(pdr <* product_definition_relationship | (pdr.id =
```

```
        'tool part relationship') AND ((SIZEOF(QUERY(prpc <* USEDIN(pdr.
        relating_product_definition.formation.of_product,'AUTOMOTIVE_DESIGN.'
        + 'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | prpc.name = 'tool'))
        < 1) OR (SIZEOF(QUERY(prpc <* USEDIN(pdr.related_product_definition.
        formation.of_product,'AUTOMOTIVE_DESIGN.' +
        'PRODUCT_RELATED_PRODUCT_CATEGORY.PRODUCTS') | prpc.name = 'part')) <
        1)))) = 0;
END_RULE;


RULE restrict_product_concept_features_for_product_specification FOR (
     configurable_item);
 WHERE
  wr1 : SIZEOF(QUERY(ci <* configurable_item | (ci.description =
        'product specification') AND (SIZEOF(ci.item_concept_feature) <> (
        SIZEOF(QUERY(pcfa <* ci.item_concept_feature | (ci.item_concept :=:
        pcfa.concept)))) + (SIZEOF(QUERY(pcfa <* ci.item_concept_feature | (ci
        .item_concept :<>: pcfa.concept) AND (SIZEOF(QUERY(pc <*
        product_class_hierarchy(pcfa.concept,[pcfa.concept]) | pc :=: ci.
        item_concept)) > 0))))))) = 0;
END_RULE;


RULE restrict_version_assignment_for_action_directive FOR (action_directive);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* action_directive | SIZEOF(USEDIN(ent,
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_version_assignment_for_configurable_item FOR (configurable_item)
                                                                  ;
 WHERE
  wr1 : SIZEOF(QUERY(ent <* configurable_item | SIZEOF(USEDIN(ent,
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_version_assignment_for_document FOR (document);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* document | SIZEOF(USEDIN(ent,
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_version_assignment_for_draughting_model FOR (draughting_model);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* draughting_model | SIZEOF(USEDIN(ent,
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE
 restrict_version_assignment_for_mechanical_design_geometric_presentation_representation
```

```
     FOR (mechanical_design_geometric_presentation_representation);
 WHERE
  wr1 : SIZEOF(QUERY(ent <*
        mechanical_design_geometric_presentation_representation | SIZEOF(
        USEDIN(ent,'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_version_assignment_for_physically_modelled_product_definition FOR
     (physically_modelled_product_definition);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* physically_modelled_product_definition | SIZEOF(
        USEDIN(ent,'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_version_assignment_for_presentation_area FOR (presentation_area)
                                                                 ;
 WHERE
  wr1 : SIZEOF(QUERY(ent <* presentation_area | SIZEOF(USEDIN(ent,
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_version_assignment_for_process_plan FOR (process_plan);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* process_plan | SIZEOF(USEDIN(ent,
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_version_assignment_for_product_class FOR (product_class);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_class | SIZEOF(USEDIN(ent,
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_version_assignment_for_product_component FOR (product_component)
                                                                 ;
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_component | SIZEOF(USEDIN(ent,
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_version_assignment_for_product_concept_feature FOR (
     product_concept_feature);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_concept_feature | SIZEOF(USEDIN(ent,
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;
```

```
RULE restrict_version_assignment_for_product_definition FOR (
     product_definition);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* product_definition | SIZEOF(USEDIN(ent,
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_version_assignment_for_representation FOR (representation);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* representation | SIZEOF(USEDIN(ent,
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_version_assignment_for_shape_representation FOR (
     shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* shape_representation | SIZEOF(USEDIN(ent,
        'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE restrict_version_assignment_for_solution_based_product_definition FOR (
     solution_based_product_definition);
 WHERE
  wr1 : SIZEOF(QUERY(ent <* solution_based_product_definition | SIZEOF(USEDIN(
        ent,'AUTOMOTIVE_DESIGN.VERSION_ASSIGNMENT.ITEMS')) > 1)) = 0;
END_RULE;


RULE retention_requires_retention_assignment FOR (retention);
 WHERE
  wr1 : SIZEOF(QUERY(r <* retention | SIZEOF(QUERY(aa <* USEDIN(r,
        'AUTOMOTIVE_DESIGN.' + 'ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
        'AUTOMOTIVE_DESIGN.RETENTION_ASSIGNMENT' IN TYPEOF(aa))) = 0)) = 0;
END_RULE;


RULE security_classification_requires_security_classification_assignment FOR (
     security_classification);
 WHERE
  wr1 : SIZEOF(QUERY(sc <* security_classification | SIZEOF(USEDIN(sc,
        'AUTOMOTIVE_DESIGN.SECURITY_CLASSIFICATION_ASSIGNMENT.' +
        'ASSIGNED_SECURITY_CLASSIFICATION')) = 0)) = 0;
END_RULE;


RULE shape_aspect_relationship_subtype_exclusiveness FOR (
     shape_aspect_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(sar <* shape_aspect_relationship | (SIZEOF(TYPEOF(sar) *
        ['AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP',
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP',
```

```
        'AUTOMOTIVE_DESIGN.SHAPE_ASPECT_DERIVING_RELATIONSHIP',
        'AUTOMOTIVE_DESIGN.DIMENSIONAL_LOCATION']) > 1) OR ((SIZEOF(TYPEOF(sar
        )) > 2) AND (SIZEOF(TYPEOF(sar) * [
        'AUTOMOTIVE_DESIGN.FEATURE_COMPONENT_RELATIONSHIP',
        'AUTOMOTIVE_DESIGN.SHAPE_DEFINING_RELATIONSHIP',
        'AUTOMOTIVE_DESIGN.SHAPE_ASPECT_DERIVING_RELATIONSHIP',
        'AUTOMOTIVE_DESIGN.DIMENSIONAL_LOCATION']) = 1)))) = 0;
END_RULE;


RULE shape_aspect_subtype_exclusiveness FOR (shape_aspect);
 WHERE
  wr1 : SIZEOF(QUERY(sa <* shape_aspect | (SIZEOF(TYPEOF(sa) * [
        'AUTOMOTIVE_DESIGN.BEAD_END','AUTOMOTIVE_DESIGN.BOSS_TOP',
        'AUTOMOTIVE_DESIGN.CHAMFER_OFFSET',
        'AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.CLOSED_PATH_PROFILE','AUTOMOTIVE_DESIGN.DATUM',
        'AUTOMOTIVE_DESIGN.DATUM_FEATURE','AUTOMOTIVE_DESIGN.DATUM_TARGET',
        'AUTOMOTIVE_DESIGN.DIRECTED_ANGLE','AUTOMOTIVE_DESIGN.HOLE_BOTTOM',
        'AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
        'AUTOMOTIVE_DESIGN.JOGGLE_TERMINATION',
        'AUTOMOTIVE_DESIGN.LINEAR_PROFILE',
        'AUTOMOTIVE_DESIGN.NGON_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.OPEN_PATH_PROFILE',
        'AUTOMOTIVE_DESIGN.PARTIAL_CIRCULAR_PROFILE',
        'AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT',
        'AUTOMOTIVE_DESIGN.PLACED_FEATURE','AUTOMOTIVE_DESIGN.POCKET_BOTTOM',
        'AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.ROUNDED_U_PROFILE','AUTOMOTIVE_DESIGN.SLOT_END',
        'AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE','AUTOMOTIVE_DESIGN.TAPER',
        'AUTOMOTIVE_DESIGN.TEE_PROFILE','AUTOMOTIVE_DESIGN.THREAD_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.TRANSITION_FEATURE','AUTOMOTIVE_DESIGN.VEE_PROFILE'
        ]) > 1) OR ((SIZEOF(TYPEOF(sa)) > 2) AND (SIZEOF(TYPEOF(sa) * [
        'AUTOMOTIVE_DESIGN.BEAD_END','AUTOMOTIVE_DESIGN.BOSS_TOP',
        'AUTOMOTIVE_DESIGN.CHAMFER_OFFSET',
        'AUTOMOTIVE_DESIGN.CIRCULAR_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.CLOSED_PATH_PROFILE','AUTOMOTIVE_DESIGN.DATUM',
        'AUTOMOTIVE_DESIGN.DATUM_FEATURE','AUTOMOTIVE_DESIGN.DATUM_TARGET',
        'AUTOMOTIVE_DESIGN.DIRECTED_ANGLE','AUTOMOTIVE_DESIGN.HOLE_BOTTOM',
        'AUTOMOTIVE_DESIGN.INSTANCED_FEATURE',
        'AUTOMOTIVE_DESIGN.JOGGLE_TERMINATION',
        'AUTOMOTIVE_DESIGN.LINEAR_PROFILE',
        'AUTOMOTIVE_DESIGN.NGON_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.OPEN_PATH_PROFILE',
        'AUTOMOTIVE_DESIGN.PARTIAL_CIRCULAR_PROFILE',
        'AUTOMOTIVE_DESIGN.PATH_FEATURE_COMPONENT',
        'AUTOMOTIVE_DESIGN.PLACED_FEATURE','AUTOMOTIVE_DESIGN.POCKET_BOTTOM',
        'AUTOMOTIVE_DESIGN.RECTANGULAR_CLOSED_PROFILE',
        'AUTOMOTIVE_DESIGN.ROUNDED_U_PROFILE','AUTOMOTIVE_DESIGN.SLOT_END',
        'AUTOMOTIVE_DESIGN.SQUARE_U_PROFILE','AUTOMOTIVE_DESIGN.TAPER',
        'AUTOMOTIVE_DESIGN.TEE_PROFILE','AUTOMOTIVE_DESIGN.THREAD_OCCURRENCE',
        'AUTOMOTIVE_DESIGN.TRANSITION_FEATURE','AUTOMOTIVE_DESIGN.VEE_PROFILE'
        ]) = 1)))) = 0;
```

```
END_RULE;


RULE shape_representation_subtype_exclusiveness FOR (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(pd <* shape_representation | (SIZEOF(TYPEOF(pd) * [
        'AUTOMOTIVE_DESIGN.ADVANCED_BREP_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.COMPOUND_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.CSG_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.FACETED_BREP_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.FEATURED_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION'
        ,
  'AUTOMOTIVE_DESIGN.GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.HYBRID_3D_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.MANIFOLD_SURFACE_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.NON_MANIFOLD_SURFACE_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.POINT_PLACEMENT_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.SHAPE_DIMENSION_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS']) > 1) OR ((
        SIZEOF(TYPEOF(pd)) > 2) AND (SIZEOF(TYPEOF(pd) * [
        'AUTOMOTIVE_DESIGN.ADVANCED_BREP_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.COMPOUND_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.CSG_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.FACETED_BREP_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.FEATURED_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION'
        ,
  'AUTOMOTIVE_DESIGN.GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.HYBRID_3D_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.MANIFOLD_SURFACE_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.NON_MANIFOLD_SURFACE_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.POINT_PLACEMENT_SHAPE_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.SHAPE_DIMENSION_REPRESENTATION',
        'AUTOMOTIVE_DESIGN.SHAPE_REPRESENTATION_WITH_PARAMETERS']) = 1)))) = 0
        ;
END_RULE;


RULE sheets_belong_to_one_drawing FOR (drawing_sheet_revision,
                                       drawing_sheet_revision_usage);
 WHERE
  wr1 : SIZEOF(QUERY(dsr <* drawing_sheet_revision | NOT (SIZEOF(QUERY(usage <*
        drawing_sheet_revision_usage | (dsr :=: usage.area))) = 1))) = 0;
END_RULE;


RULE styled_curve FOR (styled_item);
 WHERE
```

```
  wr1 : SIZEOF(QUERY(crv <* QUERY(si <* styled_item | (
        'AUTOMOTIVE_DESIGN.CURVE' IN TYPEOF(si.item))) | NOT (SIZEOF(QUERY(sty
        <* crv.styles | NOT ((SIZEOF(sty.styles) = 1) AND (
        'AUTOMOTIVE_DESIGN.CURVE_STYLE' IN TYPEOF(sty.styles[1])))))) = 0))) =
        0;
END_RULE;


RULE subtype_mandatory_camera_image_2d_or_3d FOR (camera_image);
 WHERE
  wr1 : SIZEOF(QUERY(ci <* camera_image | NOT (SIZEOF(TYPEOF(ci) * [
        'AUTOMOTIVE_DESIGN.CAMERA_IMAGE_2D_WITH_SCALE',
        'AUTOMOTIVE_DESIGN.CAMERA_IMAGE_3D_WITH_SCALE']) = 1))) = 0;
END_RULE;


RULE subtype_mandatory_camera_model_2d_or_3d FOR (camera_model);
 WHERE
  wr1 : SIZEOF(QUERY(cm <* camera_model | NOT (SIZEOF(TYPEOF(cm) * [
        'AUTOMOTIVE_DESIGN.CAMERA_MODEL_D2',
        'AUTOMOTIVE_DESIGN.CAMERA_MODEL_D3']) = 1))) = 0;
END_RULE;


RULE subtype_mandatory_drawing_revision FOR (drawing_revision);
 WHERE
  wr1 : SIZEOF(QUERY(dr <* drawing_revision | NOT (
        'AUTOMOTIVE_DESIGN.AUTO_DESIGN_DRAWING_REVISION' IN TYPEOF(dr)))) = 0;
END_RULE;


RULE subtype_mandatory_effectivity FOR (effectivity);
 WHERE
  wr1 : SIZEOF(QUERY(e <* effectivity | ((SIZEOF(TYPEOF(e)) <= 2) AND (SIZEOF(
        TYPEOF(e) * ['AUTOMOTIVE_DESIGN.SERIAL_NUMBERED_EFFECTIVITY',
        'AUTOMOTIVE_DESIGN.LOT_EFFECTIVITY',
        'AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY',
        'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY']) <> 1)) OR ((SIZEOF(
        TYPEOF(e)) = 3) AND (SIZEOF(TYPEOF(e) * [
        'AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY',
        'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY']) <> 2)) OR (SIZEOF(TYPEOF
        (e)) > 3))) = 0;
  wr2 : SIZEOF(QUERY(e <* effectivity | ((SIZEOF(TYPEOF(e)) <= 2) AND (SIZEOF(
        TYPEOF(e) * ['AUTOMOTIVE_DESIGN.SERIAL_NUMBERED_EFFECTIVITY',
        'AUTOMOTIVE_DESIGN.LOT_EFFECTIVITY',
        'AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY',
        'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY']) <> 1)) OR ((SIZEOF(
        TYPEOF(e)) = 3) AND (SIZEOF(TYPEOF(e) * [
        'AUTOMOTIVE_DESIGN.DATED_EFFECTIVITY',
        'AUTOMOTIVE_DESIGN.EVENT_BASED_EFFECTIVITY']) <> 2)) OR (SIZEOF(TYPEOF
        (e)) > 3))) = 0;
END_RULE;
```

```
RULE subtype_mandatory_pre_defined_colour FOR (pre_defined_colour);
 WHERE
  wr1 : SIZEOF(QUERY(pdc <* pre_defined_colour | NOT (
        'AUTOMOTIVE_DESIGN.DRAUGHTING_PRE_DEFINED_COLOUR' IN TYPEOF(pdc)))) =
        0;
END_RULE;


RULE subtype_mandatory_pre_defined_curve_font FOR (pre_defined_curve_font);
 WHERE
  wr1 : SIZEOF(QUERY(pdcf <* pre_defined_curve_font | NOT (
        'AUTOMOTIVE_DESIGN.DRAUGHTING_PRE_DEFINED_CURVE_FONT' IN TYPEOF(pdcf))
        )) = 0;
END_RULE;


RULE subtype_mandatory_pre_defined_text_font FOR (pre_defined_text_font);
 WHERE
  wr1 : SIZEOF(QUERY(pdtf <* pre_defined_text_font | NOT (
        'AUTOMOTIVE_DESIGN.DRAUGHTING_PRE_DEFINED_TEXT_FONT' IN TYPEOF(pdtf)))
        ) = 0;
END_RULE;


RULE subtype_mandatory_product_concept FOR (product_concept);
 WHERE
  wr1 : SIZEOF(QUERY(pc <* product_concept | (SIZEOF(TYPEOF(pc)) = 1) OR (
        SIZEOF(TYPEOF(pc) * ['AUTOMOTIVE_DESIGN.PRODUCT_CLASS']) = 0))) = 0;
END_RULE;


RULE subtype_mandatory_product_definition_with_associated_documents FOR (
     product_definition_with_associated_documents);
 WHERE
  wr1 : SIZEOF(QUERY(pdwad <* product_definition_with_associated_documents | (
        SIZEOF(TYPEOF(pdwad)) = 1) OR (SIZEOF(TYPEOF(pdwad) * [
        'AUTOMOTIVE_DESIGN.PHYSICALLY_MODELLED_PRODUCT_DEFINITION']) = 0))) =
        0;
END_RULE;


RULE terminator_symbol_constraint FOR (terminator_symbol);
 WHERE
  wr1 : SIZEOF(QUERY(ts <* terminator_symbol | ((
        'AUTOMOTIVE_DESIGN.DEFINED_SYMBOL' IN TYPEOF(ts.item)) AND NOT (
        'AUTOMOTIVE_DESIGN.PRE_DEFINED_TERMINATOR_SYMBOL' IN TYPEOF(ts.item\
        defined_symbol.definition))))) = 0;
END_RULE;


RULE text_font_usage FOR (externally_defined_text_font,pre_defined_text_font);
 WHERE
  wr1 : SIZEOF(QUERY(pdtf <* pre_defined_text_font | NOT (SIZEOF(USEDIN(pdtf,
        'AUTOMOTIVE_DESIGN.' + 'TEXT_LITERAL.FONT')) >= 1))) = 0;
```

```
   wr2 : SIZEOF(QUERY(edtf <* externally_defined_text_font | NOT (SIZEOF(USEDIN
         (edtf,'AUTOMOTIVE_DESIGN.' + 'TEXT_LITERAL.FONT')) >= 1))) = 0;
 END_RULE;


 RULE versioned_action_request_requires_date_and_person_or_organization FOR (
      versioned_action_request);
  WHERE
   wr1 : SIZEOF(QUERY(v_a_r <* versioned_action_request | (SIZEOF(QUERY(adapa <*
         USEDIN(v_a_r,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_DATE_AND_PERSON_ASSIGNMENT.ITEMS') | adapa.role.name =
         'notified person or organization')) + SIZEOF(QUERY(adapaoa <* USEDIN(
         v_a_r,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_DATE_AND_PERSON_AND_ORGANIZATION_ASSIGNMENT.' + 'ITEMS') |
         adapaoa.role.name = 'notified person or organization')) + SIZEOF(QUERY
         (adaoa <* USEDIN(v_a_r,'AUTOMOTIVE_DESIGN.' +
         'APPLIED_DATE_AND_ORGANIZATION_ASSIGNMENT.ITEMS') | adaoa.role.name =
         'notified person or organization'))) = 0)) = 0;
 END_RULE;

END_SCHEMA;

(*
```

Annex B
(normative)
AIM short names

Table B.1 provides the short names of entities specified in the AIM of this part of ISO 10303. Requirements on the use of the short names are found in the implementation methods included in ISO 10303.

Table B.1 – AIM short names

Entity names	Short names
ACTION	ACTION
ACTION_ASSIGNMENT	ACTASS
ACTION_DIRECTIVE	ACTDRC
ACTION_METHOD	ACTMTH
ACTION_PROPERTY	ACTPRP
ACTION_PROPERTY_REPRESENTATION	ACPRRP
ACTION_RELATIONSHIP	ACTRLT
ACTION_REQUEST_ASSIGNMENT	ACRQAS
ACTION_REQUEST_ROLE	ACRQRL
ACTION_REQUEST_SOLUTION	ACRQSL
ACTION_REQUEST_STATUS	ACRQST
ACTION_RESOURCE	ACTRSR
ACTION_RESOURCE_REQUIREMENT	ACRSRQ
ACTION_RESOURCE_TYPE	ACRSTY
ACTION_ROLE	ACTRL
ACTION_STATUS	ACTSTT
ADDRESS	ADDRSS
ADVANCED_BREP_SHAPE_REPRESENTATION	ABSR
ADVANCED_FACE	ADVFC
ALTERNATE_PRODUCT_RELATIONSHIP	ALPRRL

Table B.1 – (continued)

Entity names	Short names
AMOUNT_OF_SUBSTANCE_UNIT	AOSU
ANGULARITY_TOLERANCE	ANGTLR
ANGULAR_DIMENSION	ANGDMN
ANGULAR_LOCATION	ANGLCT
ANGULAR_SIZE	ANGSZ
ANNOTATION_CURVE_OCCURRENCE	ANCROC
ANNOTATION_FILL_AREA	ANFLAR
ANNOTATION_FILL_AREA_OCCURRENCE	AFAO
ANNOTATION_OCCURRENCE	ANNOCC
ANNOTATION_OCCURRENCE_ASSOCIATIVITY	ANOCAS
ANNOTATION_OCCURRENCE_RELATIONSHIP	ANOCRL
ANNOTATION_SUBFIGURE_OCCURRENCE	ANSBOC
ANNOTATION_SYMBOL	ANNSYM
ANNOTATION_SYMBOL_OCCURRENCE	ANSYOC
ANNOTATION_TEXT	ANNTXT
ANNOTATION_TEXT_OCCURRENCE	ANTXOC
APEX	APEX
APPLICATION_CONTEXT	APPCNT
APPLICATION_CONTEXT_ELEMENT	APCNEL
APPLICATION_CONTEXT_ELEMENT_RELATIONSHIP	ACER
APPLICATION_PROTOCOL_DEFINITION	APPRDF
APPLIED_ACTION_ASSIGNMENT	APACAS
APPLIED_ACTION_REQUEST_ASSIGNMENT	AARA
APPLIED_APPROVAL_ASSIGNMENT	APAPAS
APPLIED_CERTIFICATION_ASSIGNMENT	APCRAS

Table B.1 – (continued)

Entity names	Short names
APPLIED_DATE_AND_ORGANIZATION_ASSIGNMENT	ADAOA
APPLIED_DATE_AND_PERSON_AND_ORGANIZATION_ASSIGNMENT	ADAPAO
APPLIED_DATE_AND_PERSON_ASSIGNMENT	ADAPA
APPLIED_DATE_AND_TIME_ASSIGNMENT	ADATA
APPLIED_DATE_ASSIGNMENT	APDTAS
APPLIED_DOCUMENT_REFERENCE	APDCRF
APPLIED_EFFECTIVE_CONTEXT_ASSIGNMENT	AECA
APPLIED_EFFECTIVE_ITEM_ASSIGNMENT	AEI0
APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT	AEIA
APPLIED_GROUP_ASSIGNMENT	APGRAS
APPLIED_IDENTIFICATION_ASSIGNMENT	APIDAS
APPLIED_MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT	AMLAA
APPLIED_ORGANIZATION_ASSIGNMENT	APORAS
APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT	APAOA
APPLIED_PERSON_ASSIGNMENT	APPRAS
APPLIED_PRESENTED_ITEM	APPRIT
APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT	ASCA
APPROVAL	APPRVL
APPROVAL_ASSIGNMENT	APPASS
APPROVAL_DATE_TIME	APDTTM
APPROVAL_DATE_TIME_ROLE	ADTR
APPROVAL_PERSON_ORGANIZATION	APPROR
APPROVAL_RELATIONSHIP	APPRLT
APPROVAL_ROLE	APPRL
APPROVAL_STATUS	APPSTT

Table B.1 – (continued)

Entity names	Short names
AREA_IN_SET	ARINST
AREA_MEASURE_WITH_UNIT	AMWU
AREA_PRESENTATION_RELATIONSHIP	ARPRRL
AREA_UNIT	ARUNT
ASSEMBLY_COMPONENT_USAGE	ASCMUS
ASSEMBLY_COMPONENT_USAGE_SUBSTITUTE	ACUS
AUTO_DESIGN_DRAWING_REVISION	ADDR
AXIS1_PLACEMENT	AX1PLC
AXIS2_PLACEMENT_2D	A2PL2D
AXIS2_PLACEMENT_3D	A2PL3D
BACKGROUND_COLOUR	BCKCLR
BEAD	BEAD
BEAD_END	BDEND
BEZIER_CURVE	BZRCRV
BEZIER_SURFACE	BZRSRF
BLOCK	BLOCK
BOOLEAN_RESULT	BLNRSL
BOSS	BOSS
BOSS_TOP	BSSTP
BOUNDARY_CURVE	BNDCR
BOUNDED_CURVE	BNDCRV
BOUNDED_SURFACE	BNDSRF
BOXED_HALF_SPACE	BXHLSP
BOX_DOMAIN	BXDMN
BREP_WITH_VOIDS	BRWTVD

Table B.1 – (continued)

Entity names	Short names
B_SPLINE_CURVE_WITH_KNOTS	BSCWK
B_SPLINE_SURFACE	BSPSR
B_SPLINE_SURFACE_WITH_KNOTS	BSSWK
CALENDAR_DATE	CLNDT
CAMERA_IMAGE	CMRIMG
CAMERA_IMAGE_2D_WITH_SCALE	CI2WS
CAMERA_IMAGE_3D_WITH_SCALE	CI3WS
CAMERA_MODEL	CMRMDL
CAMERA_MODEL_D2	CMMDD2
CAMERA_MODEL_D3	CMMDD3
CAMERA_MODEL_D3_WITH_HLHSR	CMDWH
CAMERA_USAGE	CMRUSG
CARTESIAN_POINT	CRTPNT
CARTESIAN_TRANSFORMATION_OPERATOR	CRTROP
CARTESIAN_TRANSFORMATION_OPERATOR_2D	CTO2
CARTESIAN_TRANSFORMATION_OPERATOR_3D	CTO3
CENTRE_OF_SYMMETRY	CNOFSY
CERTIFICATION	CRTFCT
CERTIFICATION_ASSIGNMENT	CRTASS
CERTIFICATION_ROLE	CRTRL
CERTIFICATION_TYPE	CRTTYP
CHAMFER	CHMFR
CHAMFER_OFFSET	CHMOFF
CHARACTERIZED_OBJECT	CHROBJ
CHARACTERIZED_OBJECT_RELATIONSHIP	CHOBRL

Table B.1 – (continued)

Entity names	Short names
CIRCLE	CIRCLE
CIRCULARITY_TOLERANCE	CRCTLR
CIRCULAR_CLOSED_PROFILE	CRCLPR
CIRCULAR_PATTERN	CRCPTT
CIRCULAR_RUNOUT_TOLERANCE	CRRNTL
CLASSIFICATION_SYSTEM	CLSSYS
CLOSED_PATH_PROFILE	CLPTPR
CLOSED_SHELL	CLSSHL
COLOUR	COLOUR
COLOUR_RGB	CLRRGB
COLOUR_SPECIFICATION	CLRSPC
COMPONENT_FUNCTION_DEFINITION	CMFNDF
COMPOSITE_CURVE	CMPCRV
COMPOSITE_CURVE_ON_SURFACE	CCOS
COMPOSITE_CURVE_SEGMENT	CMCRSG
COMPOSITE_SHAPE_ASPECT	CMSHAS
COMPOSITE_TEXT	CMPTXT
COMPOSITE_TEXT_WITH_ASSOCIATED_CURVES	CTWAC
COMPOSITE_TEXT_WITH_BLANKING_BOX	CTWBB
COMPOSITE_TEXT_WITH_EXTENT	CTWE
COMPOUND_FEATURE_IN_PANEL	CFIP
COMPOUND_FEATURE_IN_SOLID	CFIS
COMPOUND_REPRESENTATION_ITEM	CMRPIT
COMPOUND_SHAPE_REPRESENTATION	CMS0
CONCENTRICITY_TOLERANCE	CNCTLR

Table B.1 – (continued)

Entity names	Short names
CONCEPT_FEATURE_RELATIONSHIP_WITH_CONDITION	CFRWC
CONDITIONAL_CONCEPT_FEATURE	CNCNFT
CONFIGURABLE_ITEM	CNF0
CONFIGURATION_DEFINITION	CNFDFN
CONFIGURATION_DESIGN	CNFDSG
CONFIGURATION_INTERPOLATION	CNFINT
CONFIGURATION_ITEM	CNFITM
CONFIGURATION_ITEM_CONTEXT	CNITCN
CONFIGURATION_ITEM_RELATIONSHIP	CNITRL
CONFIGURATION_ITEM_WITH_CONTEXT	CIWC
CONFIGURED_EFFECTIVE_CONTEXT_ASSIGNMENT	CECA
CONFIGURED_EFFECTIVE_ITEM_ASSIGNMENT	CEIA
CONIC	CONIC
CONICAL_SURFACE	CNCSRF
CONNECTED_EDGE_SET	CNEDST
CONNECTED_FACE_SET	CNFCST
CONNECTED_FACE_SHAPE_REPRESENTATION	CFSR
CONSTRUCTIVE_GEOMETRY_REPRESENTATION	CNGMRP
CONSTRUCTIVE_GEOMETRY_REPRESENTATION_RELATIONSHIP	CGRR
CONTEXT_DEPENDENT_INVISIBILITY	CNDPIN
CONTEXT_DEPENDENT_OVER_RIDING_STYLED_ITEM	CDORSI
CONTEXT_DEPENDENT_SHAPE_REPRESENTATION	CDSR
CONTEXT_DEPENDENT_UNIT	CNDPUN
CONTRACT	CNTRCT
CONTRACT_ASSIGNMENT	CNTASS

Table B.1 – (continued)

Entity names	Short names
CONTRACT_TYPE	CNTTYP
CONVERSION_BASED_UNIT	CNBSUN
COORDINATED_UNIVERSAL_TIME_OFFSET	CUTO
CSG_SHAPE_REPRESENTATION	CSSHRP
CSG_SOLID	CSGSLD
CURVE	CURVE
CURVE_BOUNDED_SURFACE	CRBNSR
CURVE_DIMENSION	CRVDMN
CURVE_REPLICA	CRVRPL
CURVE_STYLE	CRVSTY
CURVE_STYLE_FONT	CRSTFN
CURVE_STYLE_FONT_PATTERN	CSFP
CURVE_STYLE_RENDERING	CRSTRN
CYLINDRICAL_PAIR	CYLPR
CYLINDRICAL_PAIR_RANGE	CYPRRN
CYLINDRICAL_PAIR_VALUE	CYPRVL
CYLINDRICAL_SURFACE	CYLSRF
CYLINDRICITY_TOLERANCE	CYLTLR
DATA_ENVIRONMENT	DTENV
DATE	DATE
DATED_EFFECTIVITY	DTDEFF
DATE_AND_TIME	DTANTM
DATE_AND_TIME_ASSIGNMENT	DATA
DATE_ASSIGNMENT	DTASS
DATE_ROLE	DTRL

Table B.1 – (continued)

Entity names	Short names
DATUM	DATUM
DATUM_FEATURE	DTMFTR
DATUM_FEATURE_CALLOUT	DTFTCL
DATUM_REFERENCE	DTMRFR
DATUM_TARGET	DTMTRG
DATUM_TARGET_CALLOUT	DTTRCL
DEFAULT_TOLERANCE_TABLE	DFTLTB
DEFAULT_TOLERANCE_TABLE_CELL	DTTC
DEFINED_SYMBOL	DFNSYM
DEFINITIONAL_REPRESENTATION	DFNRPR
DEGENERATE_PCURVE	DGNPCR
DEGENERATE_TOROIDAL_SURFACE	DGTRSR
DERIVED_SHAPE_ASPECT	DRSHAS
DERIVED_UNIT	DRVUNT
DERIVED_UNIT_ELEMENT	DRUNEL
DESCRIPTIVE_REPRESENTATION_ITEM	DSRPIT
DIAMETER_DIMENSION	DMTDMN
DIGITAL_DOCUMENT_REFERENCE	DGDCRF
DIMENSIONAL_CHARACTERISTIC_REPRESENTATION	DMCHRP
DIMENSIONAL_EXPONENTS	DMNEXP
DIMENSIONAL_LOCATION	DMNLCT
DIMENSIONAL_LOCATION_WITH_PATH	DLWP
DIMENSIONAL_SIZE	DMNSZ
DIMENSION_CALLOUT_COMPONENT_RELATIONSHIP	DCCR
DIMENSION_CALLOUT_RELATIONSHIP	DMCLRL

Table B.1 – (continued)

Entity names	Short names
DIMENSION_CURVE_DIRECTED_CALLOUT	DCDC
DIMENSION_CURVE_TERMINATOR	DMCRTR
DIMENSION_PAIR	DMNPR
DIMENSION_RELATED_TOLERANCE_ZONE_ELEMENT	DRTZE
DIMENSION_TEXT_ASSOCIATIVITY	DMTXAS
DIRECTED_ACTION	DRCACT
DIRECTED_ANGLE	DRCANG
DIRECTED_DIMENSIONAL_LOCATION	DRDMLC
DIRECTION	DRCTN
DIRECTION_SHAPE_REPRESENTATION	DRSHRP
DOCUMENT	DCMNT
DOCUMENT_REFERENCE	DCMRFR
DOCUMENT_RELATIONSHIP	DCMRLT
DOCUMENT_ROLE	DCMRL
DOCUMENT_TYPE	DCMTYP
DOCUMENT_USAGE_CONSTRAINT	DCUSCN
DRAUGHTING_ANNOTATION_OCCURRENCE	DRANOC
DRAUGHTING_CALLOUT	DRGCLL
DRAUGHTING_CALLOUT_RELATIONSHIP	DRCLRL
DRAUGHTING_ELEMENTS	DRGELM
DRAUGHTING_MODEL	DRGMDL
DRAUGHTING_PRE_DEFINED_COLOUR	DPDC
DRAUGHTING_PRE_DEFINED_CURVE_FONT	DPDCF
DRAUGHTING_PRE_DEFINED_TEXT_FONT	DPDTF
DRAUGHTING_SPECIFICATION_REFERENCE	DRSPRF

Table B.1 – (continued)

Entity names	Short names
DRAUGHTING_SYMBOL_REPRESENTATION	DRSYRP
DRAUGHTING_TEXT_LITERAL_WITH_DELINEATION	DTLWD
DRAUGHTING_TITLE	DRGTTL
DRAWING_DEFINITION	DRWDFN
DRAWING_REVISION	DRWRVS
DRAWING_REVISION_SEQUENCE	DRRVSQ
DRAWING_SHEET_REVISION	DRSHRV
DRAWING_SHEET_REVISION_USAGE	DSRU
EDGE	EDGE
EDGE_BASED_WIREFRAME_MODEL	EBWM
EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION	EBWSR
EDGE_CURVE	EDGCRV
EDGE_LOOP	EDGLP
EDGE_ROUND	EDGRND
EFFECTIVE_CONTEXT_ASSIGNMENT	EFCNAS
EFFECTIVE_ITEM_ASSIGNMENT	EFITAS
EFFECTIVITY	EFFCTV
EFFECTIVITY_ASSIGNMENT	EFFASS
EFFECTIVITY_RELATIONSHIP	EFFRLT
EFFECTIVITY_ROLE	EFFRL
ELECTRIC_CURRENT_MEASURE_WITH_UNIT	ECMWU
ELECTRIC_CURRENT_UNIT	ELCRUN
ELEMENTARY_SURFACE	ELMSRF
ELEMENT_DELIVERY	ELMDLV
ELEMENT_DELIVERY_ASSIGNMENT	ELDLAS

Table B.1 – (continued)

Entity names	Short names
EVALUATED_DEGENERATE_PCURVE	EVDGPC
EVENT_ASSIGNMENT	EVNASS
EVENT_BASED_EFFECTIVITY	EVBSEF
EVENT_CONTEXT_ASSIGNMENT	EVCNAS
EVENT_REFERENCE	EVNRFR
EXCLUSIVE_PRODUCT_CONCEPT_FEATURE_CATEGORY	EPCFC
EXECUTED_ACTION	EXCACT
EXPRESSION_CONVERSION_BASED_UNIT	ECBU
EXPRESSION_DEFINITION	EXPDFN
EXTENSION	EXTNSN
EXTERNALLY_DEFINED_CLASSIFICATION_ATTRIBUTE	EDCA
EXTERNALLY_DEFINED_CLASSIFICATION_SYSTEM	EDCS
EXTERNALLY_DEFINED_CURVE_FONT	EDCF
EXTERNALLY_DEFINED_HATCH_STYLE	EDHS
EXTERNALLY_DEFINED_ITEM	EXDFIT
EXTERNALLY_DEFINED_PRODUCT_CATEGORY	EDPC
EXTERNALLY_DEFINED_REPRESENTATION	EXDFRP
EXTERNALLY_DEFINED_SYMBOL	EXDFSY
EXTERNALLY_DEFINED_TEXT_FONT	EDTF
EXTERNALLY_DEFINED_THREAD	EXDFTH
EXTERNALLY_DEFINED_TILE_STYLE	EDTS
EXTERNAL_IDENTIFICATION_ASSIGNMENT	EXIDAS
EXTERNAL_SOURCE	EXTSRC
EXTRUDED_FACE_SOLID	EXFCSL
FACE	FACE

Table B.1 – (continued)

Entity names	Short names
FACETED_BREP_SHAPE_REPRESENTATION	FBSR
FACE_BASED_SURFACE_MODEL	FBSM
FACE_BOUND	FCBND
FACE_OUTER_BOUND	FCOTBN
FACE_SHAPE_REPRESENTATION	FCSHRP
FACE_SURFACE	FCSRF
FEATURED_SHAPE_REPRESENTATION	FTSHRP
FEATURE_COMPONENT_DEFINITION	FTCMDF
FEATURE_COMPONENT_RELATIONSHIP	FTCMRL
FEATURE_DEFINITION	FTRDFN
FEATURE_IN_PANEL	FTINPN
FEATURE_PATTERN	FTRPTT
FILLET	FILLET
FILL_AREA_STYLE	FLARST
FILL_AREA_STYLE_COLOUR	FASC
FILL_AREA_STYLE_HATCHING	FASH
FILL_AREA_STYLE_TILES	FAST
FILL_AREA_STYLE_TILE_SYMBOL_WITH_STYLE	FASTSW
FLATNESS_TOLERANCE	FLTTLR
FULLY_CONSTRAINED_PAIR	FLCNPR
FUNCTIONALLY_DEFINED_TRANSFORMATION	FNDFTR
GEAR_PAIR	GRPR
GEAR_PAIR_RANGE	GRPRRN
GEAR_PAIR_VALUE	GRPRVL
GEOMETRICALLY_BOUNDED_2D_WIREFRAME_REPRESENTATION	GB2WR

Table B.1 – (continued)

Entity names	Short names
GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION	GBWSR
GEOMETRICAL_TOLERANCE_CALLOUT	GMTLCL
GEOMETRIC_ALIGNMENT	GMTALG
GEOMETRIC_CURVE_SET	GMCRST
GEOMETRIC_INTERSECTION	GMTINT
GEOMETRIC_REPRESENTATION_CONTEXT	GMRPCN
GEOMETRIC_REPRESENTATION_ITEM	GMRPIT
GEOMETRIC_SET	GMTST
GEOMETRIC_TOLERANCE	GMTTLR
GEOMETRIC_TOLERANCE_RELATIONSHIP	GMTLRL
GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE	GTWDR
GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT	GTWDU
GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT	GC
GLOBAL_UNIT_ASSIGNED_CONTEXT	GUAC
GROUP	GROUP
GROUP_ASSIGNMENT	GRPASS
GROUP_MEMBER_ASSIGNMENT	GRMMAS
GROUP_RELATIONSHIP	GRPRLT
GROUP_ROLE	GRPRL
HALF_SPACE_SOLID	HLSPSL
HIDDEN_ELEMENT_OVER_RIDING_STYLED_ITEM	HEORSI
HOLE_BOTTOM	HLBTT
HOLE_IN_PANEL	HLINPN
HYBRID_3D_SHAPE_REPRESENTATION	H3SR
HYPERBOLA	HYPRBL

Table B.1 – (continued)

Entity names	Short names
IDENTIFICATION_ROLE	IDNRL
INITIAL_STATE	INTSTT
INSTANCED_FEATURE	INSFTR
INTERPOLATED_CONFIGURATION_SEQUENCE	INCNSQ
INTERSECTION_CURVE	INTCRV
INVISIBILITY	INVSBL
ITEM_DEFINED_TRANSFORMATION	ITDFTR
JOGGLE	JOGGLE
JOGGLE_TERMINATION	JGGTRM
KINEMATIC_ANALYSIS_RESULT	KNANRS
KINEMATIC_FRAME_BACKGROUND_REPRESENTATION	KFBR
KINEMATIC_FRAME_BACKGROUND_REPRESENTATION_ASSOCIATION	KFBRA
KINEMATIC_FRAME_BASED_TRANSFORMATION	KFBT
KINEMATIC_GROUND_REPRESENTATION	KNGRRP
KINEMATIC_JOINT	KNMJNT
KINEMATIC_LINK	KNMLNK
KINEMATIC_LINK_REPRESENTATION	KNLNRP
KINEMATIC_LINK_REPRESENTATION_ASSOCIATION	KLRA
KINEMATIC_LINK_REPRESENTATION_RELATION	KLRR
KINEMATIC_PAIR	KNMPR
KINEMATIC_PROPERTY_DEFINITION	KNPRDF
KINEMATIC_PROPERTY_REPRESENTATION_RELATION	KPRR
KINEMATIC_STRUCTURE	KNMSTR
KNOWN_SOURCE	KNWSRC
LANGUAGE_ROLE	LNGRL

Table B.1 – (continued)

Entity names	Short names
LEADER_DIRECTED_CALLOUT	LDDRCL
LEADER_DIRECTED_DIMENSION	LDDRDM
LEADER_TERMINATOR	LDRTRM
LENGTH_MEASURE_WITH_UNIT	LMWU
LENGTH_UNIT	LNGUNT
LIGHT_SOURCE	LGHSRC
LIGHT_SOURCE_AMBIENT	LGSRAM
LIGHT_SOURCE_DIRECTIONAL	LGSRDR
LIGHT_SOURCE_POSITIONAL	LGSRPS
LIGHT_SOURCE_SPOT	LGSRSP
LIMITS_AND_FITS	LMANFT
LINE	LINE
LINEAR_DIMENSION	LNRDMN
LINEAR_PROFILE_TOLERANCE	LNPRTL
LOCAL_TIME	LCLTM
LOCATION_SHAPE_REPRESENTATION	LCSHRP
LOCATOR	LCTR
LOOP	LOOP
LOT_EFFECTIVITY	LTEFF
LUMINOUS_INTENSITY_MEASURE_WITH_UNIT	LIMWU
LUMINOUS_INTENSITY_UNIT	LMINUN
MAKE_FROM_USAGE_OPTION	MFUO
MANIFOLD_SOLID_BREP	MNSLBR
MANIFOLD_SURFACE_SHAPE_REPRESENTATION	MSSR
MAPPED_ITEM	MPPITM

Table B.1 – (continued)

Entity names	Short names
MASS_UNIT	MSSUNT
MATED_ITEM_RELATIONSHIP	MTITRL
MATERIAL_DESIGNATION	MTRDSG
MATERIAL_DESIGNATION_CHARACTERIZATION	MTDSCH
MATERIAL_PROPERTY	MTRPRP
MATERIAL_PROPERTY_REPRESENTATION	MTPRRP
MEASURE_QUALIFICATION	MSRQLF
MEASURE_REPRESENTATION_ITEM	MSRPIT
MEASURE_WITH_UNIT	MSWTUN
MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_AREA	MDGPA
MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION	MDGPR
MECHANISM	MCHNSM
MECHANISM_BASE_PLACEMENT	MCBSPL
MODEL_CHANGE	MDLCHN
MODIFIED_GEOMETRIC_TOLERANCE	MDGMTL
MODIFIED_PATTERN	MDFPTT
MULTI_LANGUAGE_ATTRIBUTE_ASSIGNMENT	MLAA
NAMED_UNIT	NMDUNT
NEXT_ASSEMBLY_USAGE_OCCURRENCE	NAUO
NGON_CLOSED_PROFILE	NGCLPR
NON_MANIFOLD_SURFACE_SHAPE_REPRESENTATION	NMSSR
OCCURRENCE_USAGE_DEFINITION	OCUSDF
OFFSET_CURVE_2D	OFCR2D
OFFSET_CURVE_3D	OFCR3D
OFFSET_SURFACE	OFFSRF

Table B.1 – (continued)

Entity names	Short names
OPEN_PATH_PROFILE	OPPTPR
OPEN_SHELL	OPNSHL
ORDINATE_DIMENSION	ORDDMN
ORGANIZATION	ORGNZT
ORGANIZATIONAL_ADDRESS	ORGADD
ORGANIZATION_ASSIGNMENT	ORGASS
ORGANIZATION_RELATIONSHIP	ORGRLT
ORGANIZATION_ROLE	ORGRL
ORIENTED_CLOSED_SHELL	ORCLSH
ORIENTED_EDGE	ORNEDG
ORIENTED_FACE	ORNFC
ORIENTED_OPEN_SHELL	OROPSH
ORIENTED_PATH	ORNPTH
OUTER_BOUNDARY_CURVE	OTBNCR
OVER_RIDING_STYLED_ITEM	ORSI
PACKAGE_PRODUCT_CONCEPT_FEATURE	PPCF
PAIR_ACTUATOR	PRACT
PAIR_VALUE	PRVL
PARABOLA	PRBL
PARALLELISM_TOLERANCE	PRLTLR
PARALLEL_OFFSET	PRLOFF
PARAMETRIC_REPRESENTATION_CONTEXT	PRRPCN
PARTIAL_CIRCULAR_PROFILE	PRCRPR
PATH	PATH
PATH_FEATURE_COMPONENT	PTFTCM

Table B.1 – (continued)

Entity names	Short names
PCURVE	PCURVE
PERPENDICULARITY_TOLERANCE	PRPTLR
PERPENDICULAR_TO	PRPT
PERSON	PERSON
PERSONAL_ADDRESS	PRSADD
PERSON_AND_ORGANIZATION	PRANOR
PERSON_AND_ORGANIZATION_ASSIGNMENT	PAOA
PERSON_AND_ORGANIZATION_ROLE	PAOR
PERSON_ASSIGNMENT	PRSASS
PERSON_ROLE	PRSRL
PHYSICALLY_MODELLED_PRODUCT_DEFINITION	PMPD
PLACED_DATUM_TARGET_FEATURE	PDT0
PLACED_FEATURE	PLCFTR
PLACEMENT	PLCMNT
PLANAR_BOX	PLNBX
PLANAR_CURVE_PAIR	PLCRPR
PLANAR_CURVE_PAIR_RANGE	PCPR
PLANAR_EXTENT	PLNEXT
PLANAR_PAIR	PLNPR
PLANAR_PAIR_RANGE	PLPRRN
PLANAR_PAIR_VALUE	PLPRVL
PLANE	PLANE
PLANE_ANGLE_MEASURE_WITH_UNIT	PAMWU
PLANE_ANGLE_UNIT	PLANUN
PLUS_MINUS_TOLERANCE	PLMNTL

Table B.1 – (continued)

Entity names	Short names
POCKET_BOTTOM	PCKBTT
POINT	POINT
POINT_ON_CURVE	PNONCR
POINT_ON_PLANAR_CURVE_PAIR	POPCP
POINT_ON_PLANAR_CURVE_PAIR_RANGE	POPCPR
POINT_ON_PLANAR_CURVE_PAIR_VALUE	POPCPV
POINT_ON_SURFACE	PNONSR
POINT_ON_SURFACE_PAIR	POSP
POINT_ON_SURFACE_PAIR_RANGE	POSPR
POINT_ON_SURFACE_PAIR_VALUE	POSPV
POINT_PLACEMENT_SHAPE_REPRESENTATION	PPSR
POINT_REPLICA	PNTRPL
POINT_STYLE	PNTSTY
POLYLINE	PLYLN
POLY_LOOP	PLYLP
POSITION_TOLERANCE	PSTTLR
PRECISION_QUALIFIER	PRCQLF
PRESENTATION_AREA	PRSAR
PRESENTATION_LAYER_ASSIGNMENT	PRLYAS
PRESENTATION_LAYER_USAGE	PRLYUS
PRESENTATION_REPRESENTATION	PRSRPR
PRESENTATION_SET	PRSST
PRESENTATION_SIZE	PRSSZ
PRESENTATION_STYLE_ASSIGNMENT	PRSTAS
PRESENTATION_STYLE_BY_CONTEXT	PSBC

Table B.1 – (continued)

Entity names	Short names
PRESENTATION_VIEW_RELATIONSHIP	PRVWRL
PRESENTED_ITEM	PRSITM
PRESENTED_ITEM_REPRESENTATION	PRITRP
PRE_DEFINED_COLOUR	PRDFCL
PRE_DEFINED_CURVE_FONT	PDCF
PRE_DEFINED_ITEM	PRDFIT
PRE_DEFINED_MARKER	PRDFMR
PRE_DEFINED_POINT_MARKER_SYMBOL	PDPMS
PRE_DEFINED_SYMBOL	PRDFSY
PRE_DEFINED_TERMINATOR_SYMBOL	PDT
PRE_DEFINED_TEXT_FONT	PDTF
PRISMATIC_PAIR	PRSPR
PRISMATIC_PAIR_RANGE	PRPRRN
PRISMATIC_PAIR_VALUE	PRPRVL
PROCESS_CHANGE	PRCCHN
PROCESS_PLAN	PRCPLN
PROCESS_PLAN_ASSIGNMENT	PRPLAS
PROCESS_PRODUCT_ASSOCIATION	PRPRAS
PROCESS_PROPERTY_ASSOCIATION	PRPRS
PRODUCT	PRDCT
PRODUCT_CATEGORY	PRDCTG
PRODUCT_CATEGORY_RELATIONSHIP	PRCTRL
PRODUCT_CLASS	PRDCLS
PRODUCT_COMPONENT	PRDCMP
PRODUCT_CONCEPT	PRDCNC

Table B.1 – (continued)

Entity names	Short names
PRODUCT_CONCEPT_FEATURE	PRCNFT
PRODUCT_CONCEPT_FEATURE_ASSOCIATION	PCFA
PRODUCT_CONCEPT_FEATURE_CATEGORY	PCF0
PRODUCT_CONCEPT_FEATURE_CATEGORY_USAGE	PCFCU
PRODUCT_CONCEPT_RELATIONSHIP	PRCNRL
PRODUCT_CONTEXT	PRDCNT
PRODUCT_DEFINITION	PRDDFN
PRODUCT_DEFINITION_CONTEXT	PRDFCN
PRODUCT_DEFINITION_FORMATION	PRDFFR
PRODUCT_DEFINITION_FORMATION_RELATIONSHIP	PDFR
PRODUCT_DEFINITION_OCCURRENCE	PRDFOC
PRODUCT_DEFINITION_PROCESS	PRDFPR
PRODUCT_DEFINITION_RELATIONSHIP	PRDFRL
PRODUCT_DEFINITION_RESOURCE	PRDFRS
PRODUCT_DEFINITION_SHAPE	PRDFSH
PRODUCT_DEFINITION_SUBSTITUTE	PRDFSB
PRODUCT_DEFINITION_USAGE	PRDFUS
PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS	PDWAD
PRODUCT_RELATED_PRODUCT_CATEGORY	PRPC
PROJECT	PRJCT
PROJECTED_ZONE_DEFINITION	PRZNDF
PROJECTION_CURVE	PRJCRV
PROJECTION_DIRECTED_CALLOUT	PRDRCL
PROJECT_ASSIGNMENT	PRJASS
PROMISSORY_USAGE_OCCURRENCE	PRUSOC

Table B.1 – (continued)

Entity names	Short names
PROPERTY_DEFINITION	PRPDFN
PROPERTY_DEFINITION_RELATIONSHIP	PRDFR
PROPERTY_DEFINITION_REPRESENTATION	PRDFRP
PROPERTY_PROCESS	PRPPRC
PROPERTY_VALUE_REPRESENTATION	PRVLRP
QUALIFIED_REPRESENTATION_ITEM	QLRPIT
QUALITATIVE_UNCERTAINTY	QLTUNC
QUANTIFIED_ASSEMBLY_COMPONENT_USAGE	QACU
QUASI_UNIFORM_CURVE	QSUNCR
QUASI_UNIFORM_SURFACE	QSUNSR
RACK_AND_PINION_PAIR	RAPP
RACK_AND_PINION_PAIR_RANGE	RAPPR
RACK_AND_PINION_PAIR_VALUE	RAPPV
RADIUS_DIMENSION	RDSDMN
RATIONAL_B_SPLINE_CURVE	RBSC
RATIONAL_B_SPLINE_SURFACE	RBSS
RATIO_MEASURE_WITH_UNIT	RMWU
RATIO_UNIT	RTUNT
RECTANGULAR_CLOSED_PROFILE	RCCLPR
RECTANGULAR_COMPOSITE_SURFACE	RCCMSR
RECTANGULAR_PATTERN	RCTPTT
RECTANGULAR_TRIMMED_SURFACE	RCTRSR
REFERENCED_MODIFIED_DATUM	RFMDDT
RELATIONSHIP_CONDITION	RLTCND
REPARAMETRISED_COMPOSITE_CURVE_SEGMENT	RCCS

Table B.1 – (continued)

Entity names	Short names
REPRESENTATION	RPRSNT
REPRESENTATION_CONTEXT	RPRCNT
REPRESENTATION_ITEM	RPRITM
REPRESENTATION_ITEM_RELATIONSHIP	RPITRL
REPRESENTATION_MAP	RPRMP
REPRESENTATION_RELATIONSHIP	RPRRLT
REPRESENTATION_RELATIONSHIP_WITH_TRANSFORMATION	RRWT
REP_ITEM_GROUP	RPITGR
REQUIREMENT_FOR_ACTION_RESOURCE	RFAR
RESOURCE_PROPERTY	RSRPRP
RESOURCE_PROPERTY_REPRESENTATION	RSPRRP
RESOURCE_REQUIREMENT_TYPE	RSRQTY
RETENTION	RTNTN
RETENTION_ASSIGNMENT	RTNASS
REVOLUTE_PAIR	RVLPR
REVOLUTE_PAIR_RANGE	RVPRRN
REVOLUTE_PAIR_VALUE	RVPRVL
REVOLVED_FACE_SOLID	RVFCSL
RIB	RIB
RIGHT_ANGULAR_WEDGE	RGANWD
RIGHT_CIRCULAR_CONE	RGCRCN
RIGHT_CIRCULAR_CYLINDER	RGCRCY
ROLLING_CURVE_PAIR	RLCRPR
ROLLING_CURVE_PAIR_VALUE	RCPV
ROLLING_SURFACE_PAIR	RLSRPR

Table B.1 – (continued)

Entity names	Short names
ROTATION_ABOUT_DIRECTION	RTABDR
ROUNDED_U_PROFILE	RNUPR
ROUND_HOLE	RNDHL
RUNOUT_ZONE_DEFINITION	RNZNDF
RUNOUT_ZONE_ORIENTATION	RNZNOR
RUNOUT_ZONE_ORIENTATION_REFERENCE_DIRECTION	RZORD
SCREW_PAIR	SCRPR
SCREW_PAIR_RANGE	SCPRRN
SCREW_PAIR_VALUE	SCPRVL
SEAM_CURVE	SMCRV
SECURITY_CLASSIFICATION	SCRCLS
SECURITY_CLASSIFICATION_ASSIGNMENT	SCCLAS
SECURITY_CLASSIFICATION_LEVEL	SCCLLV
SECURITY_CLASSIFICATION_ROLE	SCCLRL
SERIAL_NUMBERED_EFFECTIVITY	SRNMEF
SHAPE_ASPECT	SHPASP
SHAPE_ASPECT_ASSOCIATIVITY	SHASAS
SHAPE_ASPECT_DERIVING_RELATIONSHIP	SADR
SHAPE_ASPECT_RELATIONSHIP	SHASRL
SHAPE_ASPECT_TRANSITION	SHASTR
SHAPE_DEFINING_RELATIONSHIP	SHDFRL
SHAPE_DEFINITION_REPRESENTATION	SHDFRP
SHAPE_DIMENSION_REPRESENTATION	SHDMRP
SHAPE_REPRESENTATION	SHPRPR
SHAPE_REPRESENTATION_RELATIONSHIP	SHRPRL

Table B.1 – (continued)

Entity names	Short names
SHELL_BASED_SURFACE_MODEL	SBSM
SIMPLE_PAIR_RANGE	SMPRRN
SI_UNIT	SUNT
SLIDING_CURVE_PAIR	SLCRPR
SLIDING_CURVE_PAIR_VALUE	SCPV
SLIDING_SURFACE_PAIR	SLSRPR
SLIDING_SURFACE_PAIR_VALUE	SSPV
SLOT	SLOT
SLOT_END	SLTEND
SOLID_ANGLE_MEASURE_WITH_UNIT	SAMWU
SOLID_ANGLE_UNIT	SLANUN
SOLID_MODEL	SLDMDL
SOLID_REPLICA	SLDRPL
SOLUTION_ALTERNATIVE_DEFINITION	SLALDF
SOLUTION_BASED_PRODUCT_DEFINITION	SBPD
SPECIFIED_HIGHER_USAGE_OCCURRENCE	SHUO
SPHERE	SPHERE
SPHERICAL_PAIR	SPHPR
SPHERICAL_PAIR_RANGE	SPPRRN
SPHERICAL_PAIR_VALUE	SPPRVL
SPHERICAL_SURFACE	SPHSRF
SQUARE_U_PROFILE	SQUPR
STANDARD_UNCERTAINTY	STNUNC
STRAIGHTNESS_TOLERANCE	STRTLR
STRUCTURED_DIMENSION_CALLOUT	STDMCL

Table B.1 – (continued)

Entity names	Short names
SURFACE	SRFC
SURFACE_CONDITION	SRFCND
SURFACE_CONDITION_CALLOUT	SRCNCL
SURFACE_CURVE	SRFCRV
SURFACE_OF_LINEAR_EXTRUSION	SL
SURFACE_OF_REVOLUTION	SROFRV
SURFACE_PAIR	SRFPR
SURFACE_PAIR_RANGE	SRPRRN
SURFACE_PATCH	SRFPTC
SURFACE_PROFILE_TOLERANCE	SRPRTL
SURFACE_RENDERING_PROPERTIES	SRRNPR
SURFACE_REPLICA	SRFRPL
SURFACE_SIDE_STYLE	SRSDST
SURFACE_STYLE_BOUNDARY	SRSTBN
SURFACE_STYLE_CONTROL_GRID	SSCG
SURFACE_STYLE_FILL_AREA	SSFA
SURFACE_STYLE_PARAMETER_LINE	SSPL
SURFACE_STYLE_REFLECTANCE_AMBIENT	SSRA
SURFACE_STYLE_REFLECTANCE_AMBIENT_DIFFUSE	SSRAD
SURFACE_STYLE_REFLECTANCE_AMBIENT_DIFFUSE_SPECULAR	SSRADS
SURFACE_STYLE_RENDERING	SRSTRN
SURFACE_STYLE_RENDERING_WITH_PROPERTIES	SSRWP
SURFACE_STYLE_SEGMENTATION_CURVE	SSSC
SURFACE_STYLE_SILHOUETTE	SRSTSL
SURFACE_STYLE_TRANSPARENT	SRSTTR

Table B.1 – (continued)

Entity names	Short names
SWEPT_FACE_SOLID	SWFCSL
SWEPT_SURFACE	SWPSRF
SYMBOL_COLOUR	SYMCLR
SYMBOL_REPRESENTATION	SYMRPR
SYMBOL_REPRESENTATION_MAP	SYRPMP
SYMBOL_STYLE	SYMSTY
SYMBOL_TARGET	SYMTRG
SYMMETRIC_SHAPE_ASPECT	SYSHAS
SYMMETRY_TOLERANCE	SYMTLR
TANGENT	TNGNT
TAPER	TAPER
TEE_PROFILE	TPRF
TERMINATOR_SYMBOL	TRMSYM
TEXT_LITERAL	TXTLTR
TEXT_LITERAL_WITH_ASSOCIATED_CURVES	TLWAC
TEXT_LITERAL_WITH_BLANKING_BOX	TLWBB
TEXT_LITERAL_WITH_DELINEATION	TLWD
TEXT_LITERAL_WITH_EXTENT	TLWE
TEXT_STRING_REPRESENTATION	TXSTRP
TEXT_STYLE	TXTSTY
TEXT_STYLE_FOR_DEFINED_FONT	TSFDF
TEXT_STYLE_WITH_BOX_CHARACTERISTICS	TSWBC
TEXT_STYLE_WITH_MIRROR	TSWM
TEXT_STYLE_WITH_SPACING	TSWS
THERMODYNAMIC_TEMPERATURE_MEASURE_WITH_UNIT	TTMWU

Table B.1 – (continued)

Entity names	Short names
THREAD	THREAD
THREAD_OCCURRENCE	THROCC
TIME_MEASURE_WITH_UNIT	TMWU
TIME_UNIT	TMUNT
TOLERANCE_VALUE	TLRVL
TOLERANCE_ZONE	TLRZN
TOLERANCE_ZONE_DEFINITION	TLZNDF
TOLERANCE_ZONE_FORM	TLZNFR
TOPOLOGICAL_REPRESENTATION_ITEM	TPRPIT
TOROIDAL_SURFACE	TRDSRF
TORUS	TORUS
TOTAL_RUNOUT_TOLERANCE	TTRNTL
TRANSITION_FEATURE	TRNFTR
TRIMMED_CURVE	TRMCRV
TWO_DIRECTION_REPEAT_FACTOR	TDRF
TYPE_QUALIFIER	TYPQLF
UNCERTAINTY_ASSIGNED_REPRESENTATION	UNASRP
UNCERTAINTY_MEASURE_WITH_UNIT	UMWU
UNCERTAINTY_QUALIFIER	UNCQLF
UNCONSTRAINED_PAIR	UNCPR
UNCONSTRAINED_PAIR_VALUE	UNPRV
UNIFORM_CURVE	UNFCRV
UNIFORM_SURFACE	UNFSRF
UNIVERSAL_PAIR	UNVPR
UNIVERSAL_PAIR_RANGE	UNPRRN

Table B.1 – (continued)

Entity names	Short names
VECTOR	VECTOR
VECTOR_STYLE	VCTSTY
VEE_PROFILE	VPRF
VERSIONED_ACTION_REQUEST	VRACRQ
VERSION_ASSIGNMENT	VRSASS
VERTEX	VERTEX
VERTEX_LOOP	VRTLP
VERTEX_POINT	VRTPNT
VIEW_VOLUME	VWVLM
VOLUME_MEASURE_WITH_UNIT	VMWU
VOLUME_UNIT	VLMUNT

Annex C
(normative)
Implementation method-specific requirements

The implementation method defines what types of exchange behaviour are required with respect to this part of ISO 10303. Conformance to this part of ISO 10303 shall be realized in one or more implementation methods.

C.1 Requirements specific to the implementation method defined in ISO 10303–21

For an exchange structure, the file format shall be encoded according to the syntax and EXPRESS language mapping defined in ISO 10303–21 and the AIM defined in annex A of this part of ISO 10303. The header of the exchange structure shall identify use of this part of ISO 10303 by the schema name 'automotive_design' and include the object information identifier (see annex E).

The exchange structure shall not make use of the SCOPE construct (see clause 10.3 of ISO 10303–21).

Annex D
(normative)

Protocol Implementation Conformance Statement (PICS) proforma

The Protocol Implementation Conformance Statement (PICS) proforma is supplied for completion by the person or organization (the client) requesting conformance testing. Its purpose is to ascertain the scope of claimed conformance to a particular application protocol by an implementation under test (IUT) using a defined implementation method. Through the completion of this form, the PICS proforma becomes a PICS.

The information contained in the PICS is used to configure an appropriate executable test suite for use by the client.

NOTE 1 – The framework for conformance testing is given in ISO 10303–31 (Conformance testing methodology and framework: General concepts) and ISO 10303–32 (Conformance testing methodology and framework: Requirements on testing laboratories and clients). See clause 6 of this part of ISO 10303 for the conformance requirements. Refer to ISO 10303–314 (Abstract test suite: Core data for automotive mechanical design processes) for the test purposes and test cases relevant for the conformance testing of an ISO 10303–214 implementation.

The PICS proforma lists all allowed implementation options (conformance classes) and asks further questions about the implementation. These questions are related to limits of the implementation and the handling of specific situations.

The first part of the PICS proforma asks for a brief summary of the implementation's capabilities. Further parts are provided to handle the different conformance classes in more detail.

These additional parts are grouped by UoF. For each UoF, a set of application objects is identified for which additional information on their implementation is required for conformance testing. The application objects are presented in the form of a table, in which specific implementation properties are listed. The majority of questions relate to the upper limit of aggregations for attributes where no upper limits are defined in clause 4, but where in practice an upper limit is expected to be defined by an implementation. For the preprocessor, questions relate to whether an upper limit of aggregates of the specified attribute exists in the output of the implementation. For the postprocessor side, the question is whether the implementation is incapable of processing input data exceeding a fixed limit.

NOTE 2 – In cases where this limit is dependent on the underlying hardware (e.g., the amount of memory available to the process), this fact is to be described as well.

Other questions relate to the support of optional attributes, i.e., whether a preprocessor implementation will omit these attributes, or whether a postprocessor will make use of this optional information. A third group of questions is related to very specific situations, e.g., the processing of certain kinds of surface geometry, or the nesting depth for assembly structures.

With each UoF, the set of conformance classes is identified which enclose the UoF, allowing the implementor to identify the relevant tables to be filled out within the PICS.

D.1 General conformance statements

D.1.1 Protocol implementation identification

ISO 10303–214 implementation name: .....

Current version and release date: .....

D.1.2 Implementation method

Indicate the chosen implementation method and supported directions of translation.

If more than one implementation method is supported, a separate PICS proforma is to be filled out for each of them.

	ISO 10303–21	**ISO 10303–22**
Preprocessor		
Postprocessor		

D.1.3 Implemented conformance classes

NOTE – An implementation may support more than one conformance class.

	CC1	**CC2**	**CC3**	**CC4**	**CC5**	**CC6**	**CC7**	**CC8**	**CC9**	**CC10**
Preprocessor										
Postprocessor										

	CC11	**CC12**	**CC13**	**CC14**	**CC15**	**CC16**	**CC17**	**CC18**
Preprocessor								
Postprocessor								

D.2 Conformance statements for UoF C1

If conformance to class CC12, CC13, CC14, CC15, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Default_setting_–association	upper limit for default_value		
Hardness	upper limit for environment_condition		
Surface_condition	upper limit for is_applied_to		

D.3 Conformance statements for UoF D1

If conformance to class CC3, CC4, CC10, CC11, CC12, CC13, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Dimension	upper limit for composition_curve		
Drawing	upper limit for language upper limit for drawing_specification upper limit for drawing_type		

D.4 Conformance statements for UoF D2

If conformance to class CC4, CC10, CC11, CC12, CC13, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Associated_annotation_–symbol	upper limit for is_associated_to		
Associated_fill_area	upper limit for is_derived_from		

D.5 Conformance statements for UoF E1

If conformance to class CC2, CC4, CC5, CC6, CC7, CC8, CC9, CC10, CC11, CC12, CC13, CC14, CC15, CC16, CC17, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Archived_document	supported		
External_document	upper limit for language		
Physical_model	supported		

D.6 Conformance statements for UoF FF1

If conformance to class CC12, CC13, CC14, CC15, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
General_compound_feature	upper limit for elements		
General_feature	upper limit for feature_parameter		

D.7 Conformance statements for UoF FF2

If conformance to class CC12, CC13, CC14, CC15, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Circular_pattern	upper limit for relocated_feature upper limit for missing_feature		
Compound_feature_in_panel	upper limit for element		
Compound_feature_in_solid	upper limit for element		
Default_setting_–association	upper limit for default_value		
General_pattern	upper limit for added_instance		
Rectangular_pattern	upper limit for relocated_feature upper limit for missing_feature		

D.8 Conformance statements for UoF G1

If conformance to class CC3, CC4, CC10, CC11, CC12, CC13, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Constructive_geometric_–element	upper limit for geometry		
Constructive_geometry	upper limit for element upper limit for associated_model		
Curve_set_2d	upper limit for element		
Geometric_model	upper limit for additional_elements		
Wireframe_model_2d	upper limit for element		

D.9 Conformance statements for UoF G2

If conformance to class CC1, CC2, CC3, CC4, CC7, CC9, CC10, CC11, CC12, CC13, CC14, CC15, CC16, CC17, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Constructive_geometric_–element	upper limit for geometry		
Constructive_geometry	upper limit for element upper limit for associated_model		
Curve_set_3d	upper limit for element		
Geometric_model	upper limit for additional_elements		
Hybrid_geometric_model_3d	upper limit for element		
Wireframe_model_3d	upper limit for element		

D.10 Conformance statements for UoF G3

If conformance to class CC1, CC2, CC3, CC4, CC5, CC7, CC9, CC10, CC11, CC12, CC13, CC14, CC15, CC16, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Constructive_geometric_–element	upper limit for geometry		
Constructive_geometry	upper limit for element upper limit for associated_model		
Edge_loop	upper limit for edge		
Face	support of Faces with no explicit outer bound support of Faces on non-GC_1 surfaces required numerical accuracy support of Faces across seam of closed surfaces upper limit for bound		
Geometric_model	upper limit for additional_elements		
Hybrid_geometric_model_3d	upper limit for element		
Shell	upper limit for face		
Surface_model	upper limit for element		

D.11 Conformance statements for UoF G4

If conformance to class CC1, CC2, CC4, CC5, CC7, CC9, CC10, CC11, CC12, CC13, CC14, CC15, CC16, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Constructive_geometric_–element	upper limit for geometry		
Constructive_geometry	upper limit for element upper limit for associated_model		
Faceted_b_rep	upper limit for void		
Faceted_b_rep_model	upper limit for element		
Faceted_closed_shell	upper limit for shell_boundary		
Faceted_face	upper limit for bound		
Geometric_model	upper limit for additional_elements		
Hybrid_geometric_model_3d	upper limit for element		
Poly_loop	upper limit for polygon upper limit for is_bound_for		

D.12 Conformance statements for UoF G5

If conformance to class CC1, CC2, CC4, CC5, CC7, CC9, CC10, CC11, CC12, CC13, CC14, CC15, CC16, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
B_rep	upper limit for void		
B_rep_model	upper limit for element		
Closed_shell	upper limit for shell_boundary		
Constructive_geometric_–element	upper limit for geometry		
Constructive_geometry	upper limit for element upper limit for associated_model		
Edge_loop	upper limit for edge		
Face	support of Faces with no explicit outer bound support of Faces on non-GC_1 surfaces required numerical accuracy support of Faces across seam of closed surfaces upper limit for bound		
Geometric_model	upper limit for additional_elements		
Hybrid_geometric_model_3d	upper limit for element		

D.13 Conformance statements for UoF G6

If conformance to class CC5, CC11, CC12, CC13, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Compound_model	upper limit for element		
Constructive_geometric_–element	upper limit for geometry		
Constructive_geometry	upper limit for element upper limit for associated_model		
Edge_loop	upper limit for edge		
Face	support of Faces with no explicit outer bound support of Faces on non-GC_1 surfaces required numerical accuracy support of Faces across seam of closed surfaces upper limit for bound		
Geometric_model	upper limit for additional_elements		
Hybrid_geometric_model_3d	upper limit for element		
Topologically_–connected_set	upper limit for element		

D.14 Conformance statements for UoF G7

If conformance to class CC1, CC2, CC4, CC5, CC7, CC9, CC10, CC11, CC12, CC13, CC14, CC15, CC16, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
B_rep	upper limit for void		
B_rep_model	upper limit for element		
Closed_shell	upper limit for shell_boundary		
Constructive_geometric_–element	upper limit for geometry		
Constructive_geometry	upper limit for element upper limit for associated_model		
Csg_model	upper limit for tree_root_expression		
Edge_loop	upper limit for edge		
Face	support of Faces with no explicit outer bound support of Faces on non-GC_1 surfaces required numerical accuracy support of Faces across seam of closed surfaces upper limit for bound		
Faceted_b_rep	upper limit for void		
Faceted_b_rep_model	upper limit for element		
Faceted_closed_shell	upper limit for shell_boundary		
Faceted_face	upper limit for bound		
Geometric_model	upper limit for additional_elements		
Hybrid_geometric_model_3d	upper limit for element		
Poly_loop	upper limit for polygon upper limit for is_bound_for		

D.15 Conformance statements for UoF G8

If conformance to class CC1, CC2, CC4, CC7, CC9, CC10, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Constructive_geometric_–element	upper limit for geometry		
Constructive_geometry	upper limit for element upper limit for associated_model		
Geometric_model	upper limit for additional_elements		
Geometric_set_3d	upper limit for element		
Geometrically_bounded_–surface_model	upper limit for element		

D.16 Conformance statements for UoF K1

If conformance to class CC16 or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Ground_representation	upper limit for is_referenced_by		
Initial_configuration	upper limit for initial_pair_values		
Interpolated_configuration_–sequence	upper limit for interpolation		
Kinematic_configuration_–definition	upper limit for pair_values		
Kinematic_joint	upper limit for is_referenced_by		
Kinematic_link_–representation	upper limit for additional_reference_frame		
Kinematic_mechanism	upper limit for is_referenced_by		
Kinematic_structure	upper limit for joint upper limit for is_referenced_by		
Kinematic_transformation_–result	upper limit for result		
Simulation_information	upper limit for mechanism upper limit for described_object		

D.17 Conformance statements for UoF MD1

If conformance to class CC11, CC12, CC13, CC17, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Point_direction_model	upper limit for point_direction upper limit for point		
Point_direction_sequence	upper limit for element		
Point_sequence	upper limit for element		

D.18 Conformance statements for UoF PR1

If conformance to class CC6, CC7, CC8, CC9, CC10, CC11, CC12, CC13, CC14, CC15, CC16, CC17, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Default_setting_–association	upper limit for default_value		
General_item_property	upper limit for property_value		
Item_cost_property	upper limit for cost_property_value		
Item_quality_property	upper limit for quality_property_value		
Mass	upper limit for mass_property_value		
Material	upper limit for material_property		
Material_property	upper limit for material_property_value		
Material_property_–representation	upper limit for property_value		
Property_change	upper limit for added_property upper limit for deleted_property		
Recyclability	upper limit for recycling_class_value		
Thickness	upper limit for thickness_value		

D.19 Conformance statements for UoF P1

If conformance to class CC1, CC2, CC3, CC4, CC5, CC7, CC9, CC10, CC11, CC12, CC13, CC14, CC15, CC16, CC17, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Appearance_assignment	upper limit for component		
Styled_model	upper limit for element upper limit for additional_elements		
Surface_appearance	upper limit for element		

D.20 Conformance statements for UoF P2

If conformance to class CC3, CC4, CC5, CC10, CC11, CC12, CC13, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Fill_area	upper limit for boundary		

D.21 Conformance statements for UoF P3

If conformance to class CC5 or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Hidden_element_handling	upper limit for is_defined_for		
Light_source	upper limit for view		
Surface_appearance	upper limit for element		

D.22 Conformance statements for UoF S1

If conformance to class CC1, CC2, CC3, CC4, CC5, CC6, CC7, CC8, CC9, CC10, CC11, CC12, CC13, CC14, CC15, CC16, CC17, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Approval	upper limit for is_applied_to upper limit for is_approved_by		
Design_discipline_–item_definition	upper limit for is_relevant_for		
Item	upper limit for associated_version		
Item_classification	upper limit for associated_item		
Multi_language_string	upper limit for language_dependent_string		
Person	upper limit for organization		
Security_classification	upper limit for is_applied_to		

D.23 Conformance statements for UoF S2

If conformance to class CC1, CC2, CC3, CC4, CC5, CC6, CC7, CC8, CC9, CC10, CC11, CC12, CC13, CC14, CC15, CC16, CC17, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Accuracy	upper limit for is_defined_for		
Cartesian_coordinate_space	upper limit for unit		
Group	upper limit for element		
Layer	upper limit for element		
Model_change	upper limit for added_element upper limit for deleted_element		

D.24 Conformance statements for UoF S3

If conformance to class CC2, CC4, CC5, CC6, CC7, CC8, CC9, CC10, CC13, CC14, CC15, CC16, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Next_higher_assembly	supported		
Promissory_usage	supported		

D.25 Conformance statements for UoF S4

If conformance to class CC6, CC7, CC8, CC9, CC10, CC13, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Effectivity	upper limit for is_applied_to		
Retention_period	upper limit for is_applied_to		

D.26 Conformance statements for UoF S5

If conformance to class CC6, CC7, CC8, CC9, CC10, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Activity	upper limit for resolved_request upper limit for concerned_organization upper limit for supplying_organization upper limit for authorization upper limit for associated_project		
Change	upper limit for described_change		
Contract	upper limit for contracted_element upper limit for release		
Organization_in_contract	upper limit for signature upper limit for contract		
Project	upper limit for affected_product_class upper limit for work_program		
Simultaneous_activity	upper limit for element		
Work_order	upper limit for is_controlling		
Work_request	upper limit for recommended_method upper limit for notified_person upper limit for scope		

D.27 Conformance statements for UoF S6

If conformance to class CC6, CC7, CC8, CC9, CC10, CC11, CC12, CC13, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Classification_attribute	upper limit for attribute_value upper limit for associated_classification		
Classification_system	upper limit for allowed_class		
General_classification	upper limit for attribute		
Item_classification	upper limit for associated_item		

D.28 Conformance statements for UoF S7

If conformance to class CC8, CC9, CC10, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Collection_definition	upper limit for collected_items		
Design_constraint	upper limit for is_valid_for		
Final_item_solution	upper limit for final_specification		
Product_component	upper limit for is_influenced_by upper limit for is_relevant_for upper limit for is_root_entry_for		
Product_function	upper limit for is_relevant_for		
Product_specification	upper limit for specification		
Specification_expression	upper limit for operand		
Specified_higher_usage	supported		

D.29 Conformance statements for UoF S8

If conformance to class CC11, CC12, CC13, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Exclusive_process_–operation	upper limit for element		
General_process_property	upper limit for property_value		
Mated_item_relationship	upper limit for mating_material		
Mating_definition	upper limit for mated_items		
Process_cost_property	upper limit for cost_property_value		
Process_operation_–input_or_output	upper limit for concerned_shape		
Process_plan	upper limit for produced_output		
Process_quality_property	upper limit for quality_property_value		
Simultaneous_process_–operation	upper limit for element		

D.30 Conformance statements for UoF T1

If conformance to class CC4, CC10, CC12, CC13, CC14, CC15, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Default_setting_–association	upper limit for default_value		
Derived_geometry	upper limit for is_derived_from		
General_tolerance_table	upper limit for cell		

D.31 Conformance statements for UoF T2

If conformance to class CC12, CC13, CC14, CC15, or CC18 is claimed, then identify the implementation's properties with the following application objects:

Application object	Property	Preproc.	Postproc.
Compound_datum	upper limit for is_made_up_by		
Datum_target_set	upper limit for target		
Derived_geometry	upper limit for is_derived_from		
Tolerance_zone	upper limit for is_zone_for upper limit for created_by		

Annex E
(normative)
Information object registration

E.1 Document identification

To provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 10303 part(214) version(-2) }

is assigned to this part of ISO 10303. The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

E.2 Schema identification

To provide for unambiguous identification of the **automotive_design** expanded schema in an open information system, the object identifier

{ iso standard 10303 part(214) version(-2) object(1) automotive-design(1) }

is assigned to the **automotive_design** expanded schema schema (see annex A). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

To provide for unambiguous identification of the **automotive_design** short form schema in an open information system, the object identifier

{ iso standard 10303 part(214) version(-2) object(1) automotive-design(2) }

is assigned to the **automotive_design** short form schema schema (see 5.2). The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

Annex F
(informative)
Application activity model

The application activity model (AAM) is provided as an aid in understanding the scope and information requirements defined in this application protocol. The model is presented as a set of activity figures that contain the activity diagrams and a set of definitions of the activities and their data.

The AAM covers activities that go beyond the scope of this application protocol. The viewpoint of the application activity model is that of a car developer.

F.1 Application activity model definitions and abbreviations

The following terms are used in the application activity model. Terms marked with an asterisk are outside the scope of this application protocol.

The definitions given in this annex do not supersede the definitions given in the main body of the text.

F.1.1 aesthetics: Aspects of beauty, taste, and style that influence the design of a product.

F.1.2 CAD system: Computer hardware and software that supports the computer aided design of products by providing functions for defining the geometry (2D or 3D) and technological properties.

F.1.3 calculation system: Computer hardware and software that supports the determination and optimization of certain properties of the product such as strength of material.

F.1.4 car surface data: Data describing the 3D shape of the car body.

F.1.5 change request: A request for a change in a product specification in one phase of the development process generated from a different phase of the development process.

F.1.6 characteristic sections: Properties of material such as modulus of elasticity.

F.1.7 concept: Elaborated idea of a product developed on the basis of requirements based on a list of specifications.

F.1.8 conceptual product description: Preliminary geometrical description of major components of the car such as overall length, wheelbase, and gross weight.

F.1.9 constraints: Restrictions that have to be followed in order to achieve feasible results in design, simulation, or analysis.

F.1.10 corporation product strategy: The company's program for bringing the new product to market, such as advertising.

F.1.11 costs: Financial targets set by the company for developing the product or financial charges which result from product development.

F.1.12 crash test: Procedure of testing safety and behaviour of a car in the case of an accident by crashing it at a barrier or another vehicle.

F.1.13 create layout: Develop a rough packaging layout for the car as a whole including the definition of all major dimensions.

F.1.14 create physical model: Model the car body and certain components, such as wheels and instrument panel, out of wood or other material.

F.1.15 define product: Specify the essential properties of the car in the form of specification lists that is based on the given conditions such as market conditions.

F.1.16 design product: Process of defining functionality, geometry, and technological properties of the product including conceptual, preliminary, and detailed design as well as calculation.

F.1.17 design tool: Process of defining functionality, geometry and technological properties of a tool on the basis of the part description and of the production planning output.

F.1.18 develop concept: Process of elaborating initial ideas on the basis of specification lists into a detailed concept including discussions of new technologies and functional aspects of the product, possible materials and production processes.

F.1.19 develop conceptual design: Process of checking the technical concept including the design and redesign of the major parts of the car.

F.1.20 develop methods: Specification of manufacturing methods and tool properties including an initial design of the tool on the basis of production planning data and of the part description.

F.1.21 develop part design: Detailed design of the product geometry which may be the initial design or an optimization process.

F.1.22 develop product: Process of product development including definition, styling, design and evaluation of the car.

F.1.23 develop tool: Process of development of the tools needed for the production of the car; i.e. for production planning, tool design, tool manufacturing, quality control, etc.

F.1.24 develop tool design and perform analysis: Verification of the developed methods by calculations and simulations resulting in detailed descriptions of the tool geometry and of its assembly structure.

F.1.25 due dates: Deadlines defined in schedules which have to be met in the various phases of the development of the product.

F.1.26 ergonomic dummy: Physical model of a human being that is used to test the ergonomic properties of the car, such as seat, steering, and switches position.

F.1.27 ergonomics: Constraints based on the mental and physical capacities of persons.

F.1.28 evaluate product: Process of specifying, executing, and verifying tests for a prototype of a constituent or for the whole car.

F.1.29 experiences: Knowledge of developers obtained during previous product developments.

F.1.30 given quality: Requirements concerning quality set up by the company, by authorized inspection departments, or by government.

F.1.31 interpret tests: Process of generating test reports and drawing conclusions out of the test results.

F.1.32 layout description: Packaging layout of the whole car including all major dimensions.

F.1.33 load rating: Designation of an operating limit for loading used in calculation systems.

F.1.34 long term production plan: Plan of a company concerning production set up for a long period of time for aspects of manufacturing, assembling, quality assurance, and recycling including outsourcing considerations.

F.1.35 manufacture production tools: Provision of the tools which are necessary for producing the car, including NC programming and simulation.

F.1.36 manufacture prototype: Process of producing a physical prototype or a simulation model of the car or of parts of it according to a prototype specification using for example stereolithography systems.

F.1.37 manufacture tool: Process of manufacturing the tools needed for the car production, such as deep drawing dies, trimming dies, or injection moulds, using various manufacturing technologies such as milling, lathing, drilling, or laser cutting, that may be numerically controlled.

F.1.38 manufacturing equipment: Machines and tools that can be used to manufacture a product or constituents.

F.1.39 mass: Quantity of matter a body contains.

F.1.40 mass production timing: Scheduling of the distribution of tools needed for manufacturing processes and assembly lines in order to ensure just-in-time availability.

F.1.41 material description: Identification of materials and description of their properties, such as strength or endurance.

F.1.42 measured data and measurement protocols: Data obtained by 3D scanning of manufactured tools using measurement tools.

F.1.43 measurement programs: Sequences of test procedures set up to assure quality of tools and products.

F.1.44 measuring gauges: Measuring instruments having defined deviations from the nominal values for testing tolerances of dimension or shape, such as diameters of holes.

F.1.45 measuring machines: Machines that provide data about an object by measuring its properties.

F.1.46 measuring systems: Measuring machines connected to an CA-System.

F.1.47 methods plan: Technical drawing including the description of operations necessary to manufacture the part.

F.1.48 mountability: Degree of difficulty in assembling objects.

F.1.49 NC data packages: Information necessary for the NC operator to produce the part. This includes documents sent to the shop floor specifying machine setup, part loading, or tooling.

F.1.50 NC/RC programs: Sequences of statements for controlling NC machines and robots. NC/RC programs may conform with international or national standards.

F.1.51 optical impression: Effect produced in the mind of an observer at the sight of an object.

F.1.52 overall conception: Sum of abstract ideas concerning all involved aspects.

F.1.53 packaging layout system: Computer hardware and software that supports the generation of packacking layouts. A packaging layout is a plan for the arrangement of components forming a car or a part of it; this layout can exist at different levels of detail and serves the determination of an optimum solution with respect to space and ergonomics.

F.1.54 part description: Set of data or documents providing information on geometrical, functional, and technological properties of a part.

F.1.55 PDM system: Computer hardware and software that supports the computer aided management of product data by providing functions for handling regular or variational bill-of-materials and that support engineering processes such as release or change processes.

F.1.56 perform calculations: Determination of the properties and expected behaviour of the car and of its constituents using various methods, such as FEA, vibrational behaviour analysis, or strength examination in order to optimize material and dimensions and to check safety criteria.

F.1.57 plan production: Process of generating a plan for the manufacturing process of a product, including the sequence of operations.

F.1.58 plan tests: Preparation of tests including the specification of prototypes and the selection of test methods, facilities, and evaluation systems.

F.1.59 planning systems: Software tools used to organize the production process of products including the creation of process plans and the determination of manufacturing sequences.

F.1.60 preconditions: Necessary or required conditions.

F.1.61 preliminary designed components (patents): Components usable within several types of cars which are designed in a universal form and which can easily be adapted to new products in order to meet the specific requirements of a particular product (e.g., the patent of a turbo charger which can be adapted to suit a particular engine).

F.1.62 product description: All data needed to describe a product completely, including geometrical information of the shape and technological information, such as material, tolerances, and surface conditions.

F.1.63 production and assembly sequences: Results of the production planning stage laid down in process plans that specify how and in which order parts have to be manufactured and assembled.

F.1.64 production control data: Information about the production process, such as utilization rate of machines.

F.1.65 production know-how: Knowledge of how to produce something.

F.1.66 production order: Arrangement of the production process defining the way one production operation follows another.

F.1.67 program and simulate NC/RC: Generation of tool pathing using CAM systems including the definition of the NC milling strategy and the selection of milling cutters, collision detection, and simulation of the manufacturing process.

F.1.68 prototype: A model or pattern from which subsequent copies are made, or improved specimens are developed.

F.1.69 raw material: Crude unprocessed or partially processed material used as a feedstock for a processing operation.

F.1.70 RC programming system: Programming systems used to create robot control data.

F.1.71 regulations: Rules, such as special working conditions, that are set up by the government or by companies influenced by public opinion, such as aspects of environmental protection.

F.1.72 results of calculation: Values that ensue from a calculation.

F.1.73 security aspects: Conditions that define protection for persons and objects.

F.1.74 simulation systems: Systems that mimic some or all of the behaviour of one system with another system, such as die sinking simulations for the specification of manufacturing facilities.

F.1.75 sketches (characteristic sections): A rough drawing, an outline.

F.1.76 specification list: A list of something demanded as an obligation.

F.1.77 standards: Rules set up within the automotive industry or a car company in order to ensure commonality between products and tools. These can be international (e.g., ISO, IEC), national (e.g., AFNOR, ANSI, DIN), or company internal standards.

F.1.78 standard parts: Standardized parts with variable dimensions or shape. Standard parts, such as screws, are usually listed in catalogues.

F.1.79 stereolithography systems: System for the generation of 3D physical models of a part out of resin using CAD data.

F.1.80 style product: Process of specifying the shape of a car according to the specification list in consideration of aesthetical and visual aspects.

F.1.81 style shape (exterior and interior): Process of generating the data describing the 3D shape of a car including exterior and interior surfaces.

F.1.82 technical practicability: The capability of an object to perform tasks.

F.1.83 test facilities: Test sites and tools used to obtain data about a product with respect to certain characteristics, such as heat development inside the combustion chamber, wear, corrosion, and material behaviour.

F.1.84 test prototype: Process of performing tests on prototypes or of executing simulation programs as part of the product evaluation process.

F.1.85 test parts: Parts manufactured with preliminary tools used to evaluate the tools.

F.1.86 test plan data: Information on the tests to be executed on prototypes of the product in order to evaluate the product design.

F.1.87 test reports: Information output of the test stage of the product development, e.g., temperature measurements.

F.1.88 test results: Values that ensue from a test.

F.1.89 test run: Procedure of testing functional units over a certain period of time.

EXAMPLE 240 – Test run of an engine to simulate a two year wearing period within one week.

F.1.90 test tool to control quality: Manufacturing of test parts using preliminary tools in order to evaluate and optimize the tool design.

F.1.91 tolerance data: Data that gives information about a permissible deviation from a specified value.

F.1.92 tool: Piece of hardware that supports the manufacturing or assembly process of a product; e.g., die, tin-plate folder, or robot.

F.1.93 tool control: Computer sensible code containing sequences of statements necessary to control a machine tool.

F.1.94 tool database: Database containing information on existing or available tools and their attributes. This may include company specific tool data as well as catalogues from tool manufacturers.

F.1.95 tool description: All data necessary to describe a tool, including geometrical information of the shape and technological information, such as tolerances, material, and surface condition descriptions.

F.1.96 tool selection: Set of tools that fulfil certain constraints and conditions, such as costs, tool wear, and availability.

F.1.97 visualization systems: Computer hardware and software that processes product description data into a graphical form of representation in order to make easy understanding and interpretation possible.

F.2 Application activity model diagrams

The application activity model diagrams are given in figures F.1 through F.9. The graphical form of the application activity model is presented in the IDEF0 [1] activity modelling format. Activities and data flows that are out of scope are marked with asterisks.

The data flows are marked with numbers to identify the data classes which are included in these data flows. The numbers correspond to the description in table F.1.

Table F.1 consists of six columns. The first column contains the numbers of the data classes. In columns two, three, and four, the data classes are marked to be in scope (full support or reference only) or out of scope.

The classification "in scope – full support" means that corresponding definitions of application objects are provided in clause 4 of this part of ISO 10303.

The classification "in scope – reference only" means that the explicit description of the data is not specified by this part of ISO 10303, but it can be referenced.

Table F.1 – The AAM-data classes

Data class No.	In scope		Out of scope	Description	UoF correspondence
	full support	reference only			
1	X			**Drawing data:** dimensions, views, layout, etc.	D1, D2, P1, P2
2	X			**Wireframe data:** points, polygons, conics, curves, etc.	G1, G2
3	X			**Surface data:** unbounded and bounded surfaces planes, open shells, etc.	G3, G8
4	X			**Solid data:** B–Rep, CSG, etc.	G4, G5, G7
5		X		**Process Control Data:** NC–data, RC–data, MC–data, etc.	E1
6	X			**Visualization data:** color, line style, point style, etc.	P1 – P3
				Simulation data:	
7a	X			kinematics	K1
7b			X	other simulation data	—
				Compound model and analysis data:	
8a	X			compound representation	G6
8b			X	FEA data	—
9	X			**Form feature data:** general feature, solid features, panel features, transition features, compound features, patterns	FF1 – FF3
10			X	**Parametric design data**	—
11	X			**Material, Surface conditions,** **and Tolerance data**	PR1, C1 T1, T2
12	X			**Product structure and** **management data:** version and release management, work management, assembly structure, fastening information, process operation, process plan, specification control, etc.	S1 – S8
13		X		**Compound documents:** requirements, feasibility studies, etc.	E1
14a	X			**Scan data**	MD1
14b			X	**other measured data**	—

EXAMPLE 241 – The data description of a NC–data file is not specified by this part of ISO 10303. This data may be provided in an ASCII–file according to, e.g., ISO 6983 [6]. The reference to this NC–data, the knowledge about its existence and about its type, creator, etc., is supported by this part.

Data classified as "out of scope" are not supported by this part of ISO 10303. In the figures, data classes that are out of scope are marked with an asterisk after the corresponding number.

The fifth column of table F.1 contains a description of the data classes with some typical examples.

The sixth column of table F.1 lists the correspondence of the data classes to the UoF specified in clause 4.

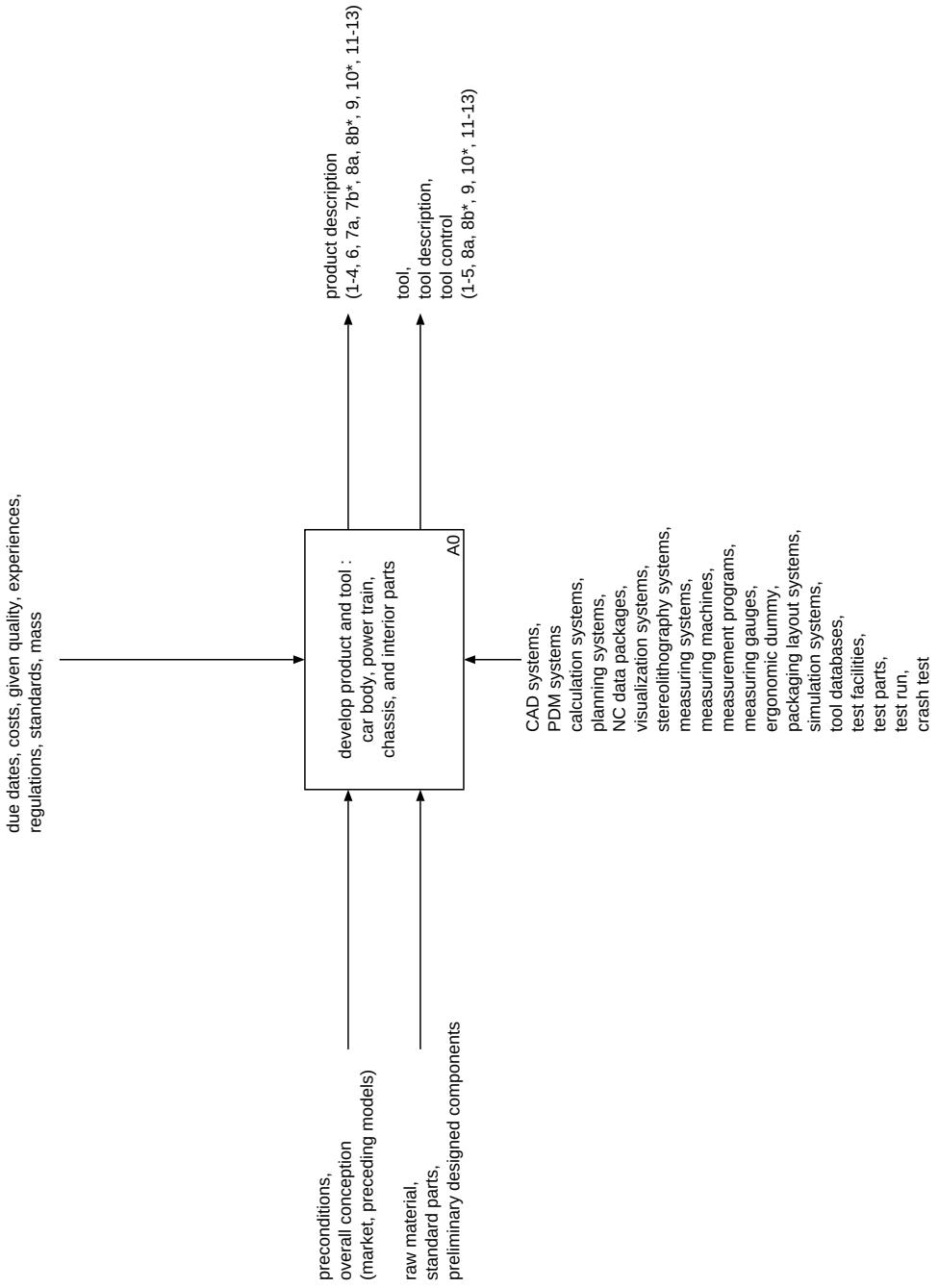


Figure F.1 – AAM diagram of node A-0: Core data for automotive mechanical design processes

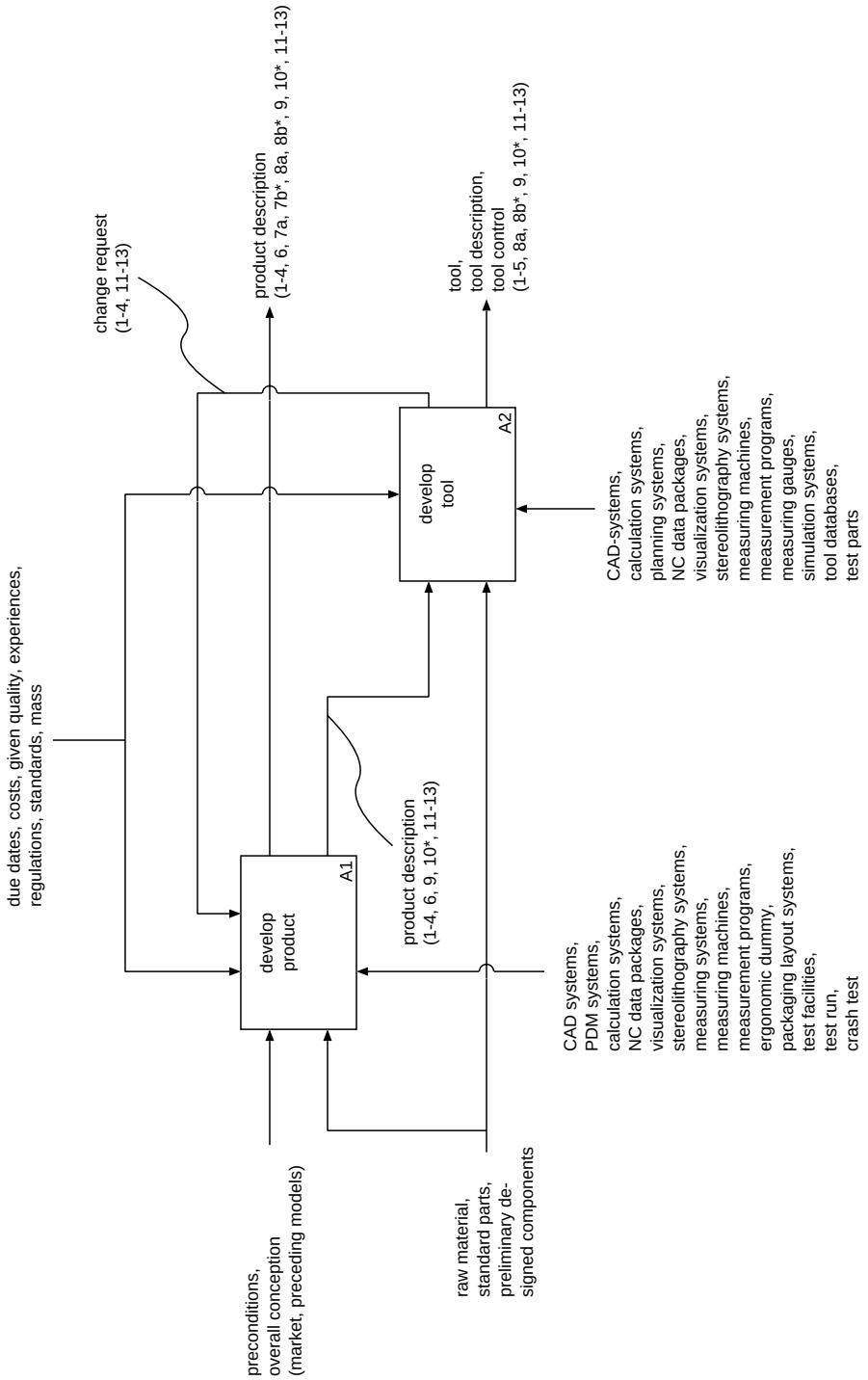


Figure F.2 – AAM diagram of node A0: Develop product and tool

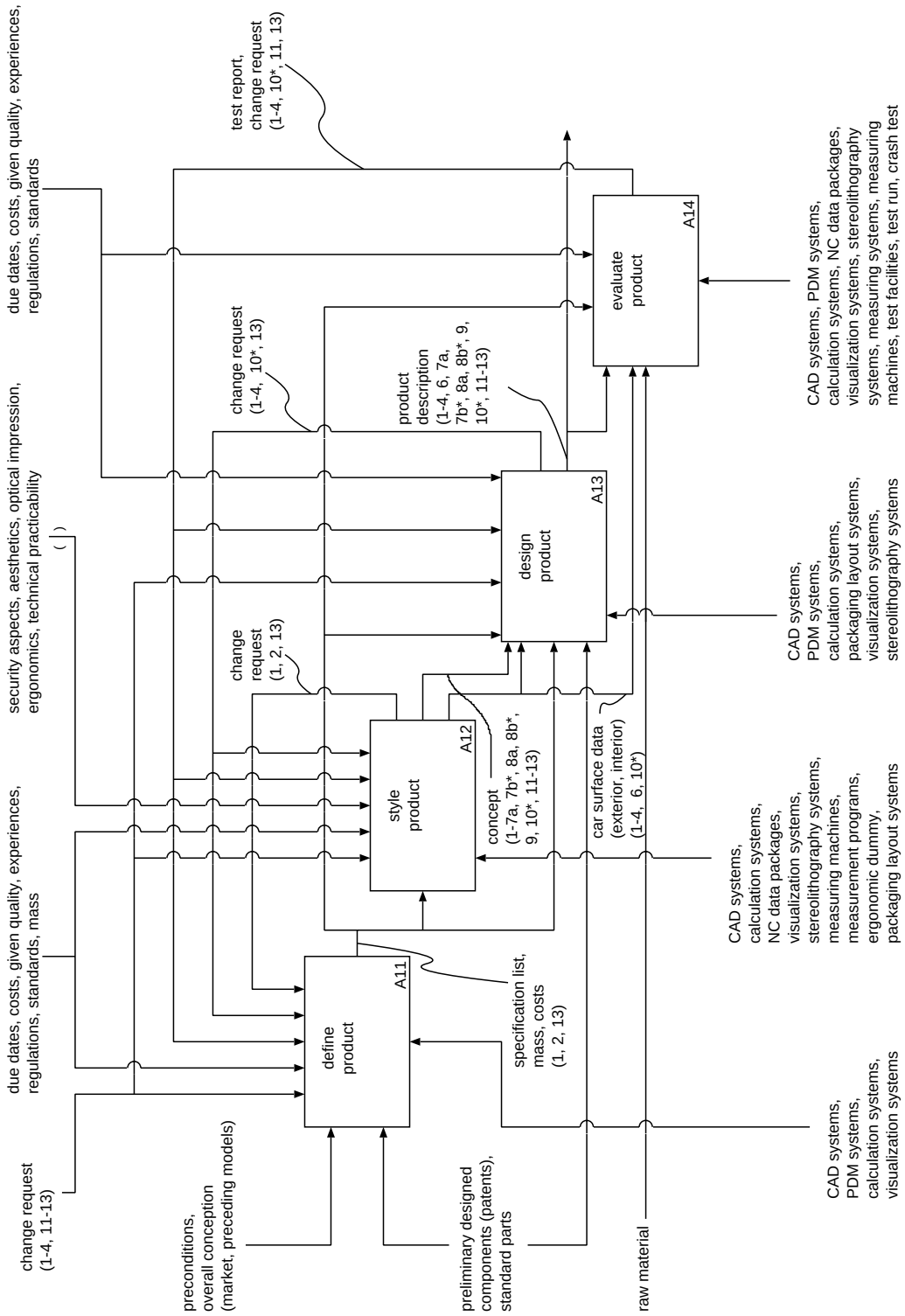


Figure F.3 – AAM diagram of node A1: Develop product

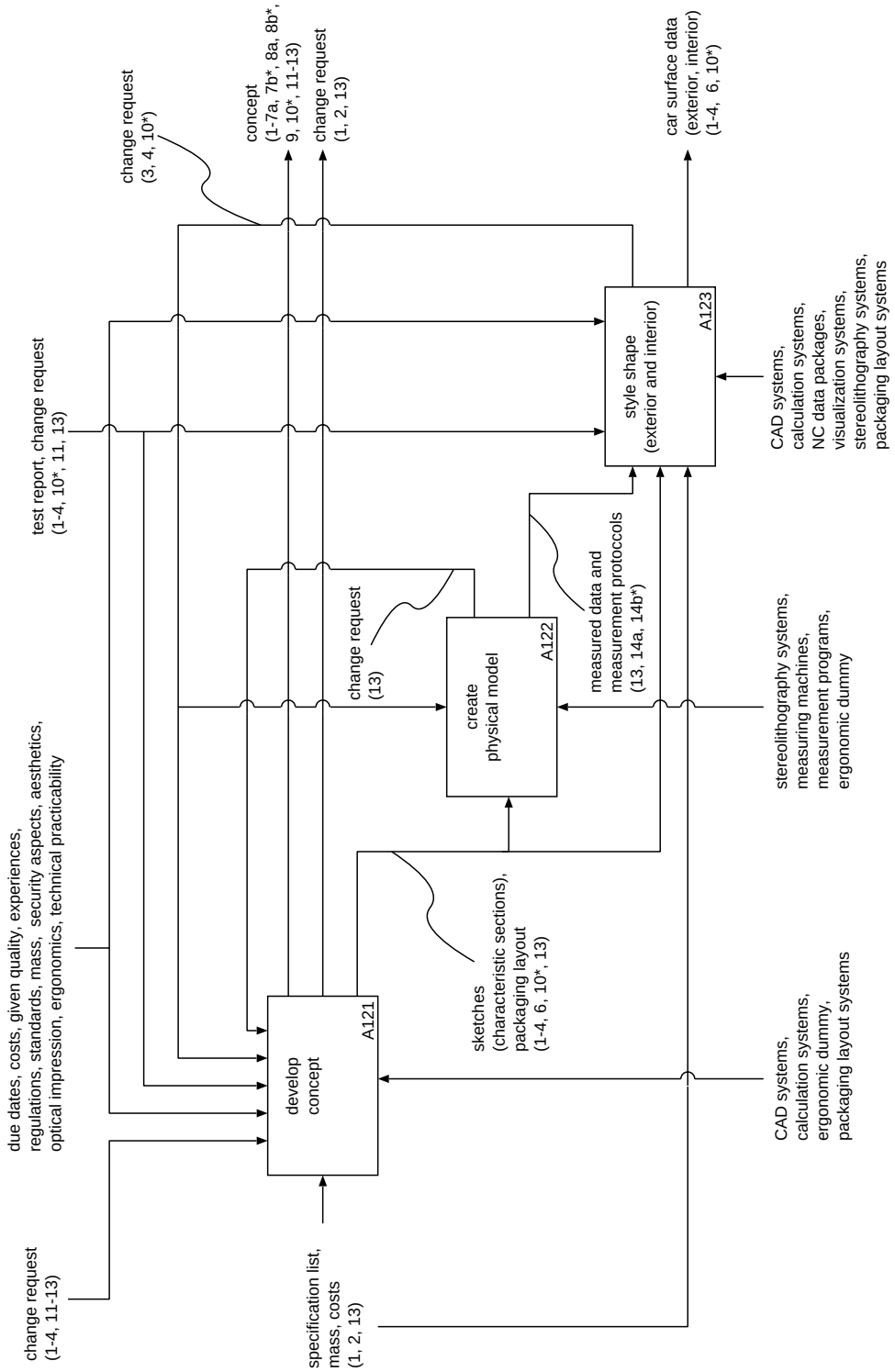


Figure F.4 – AAM diagram of node A12: Style product

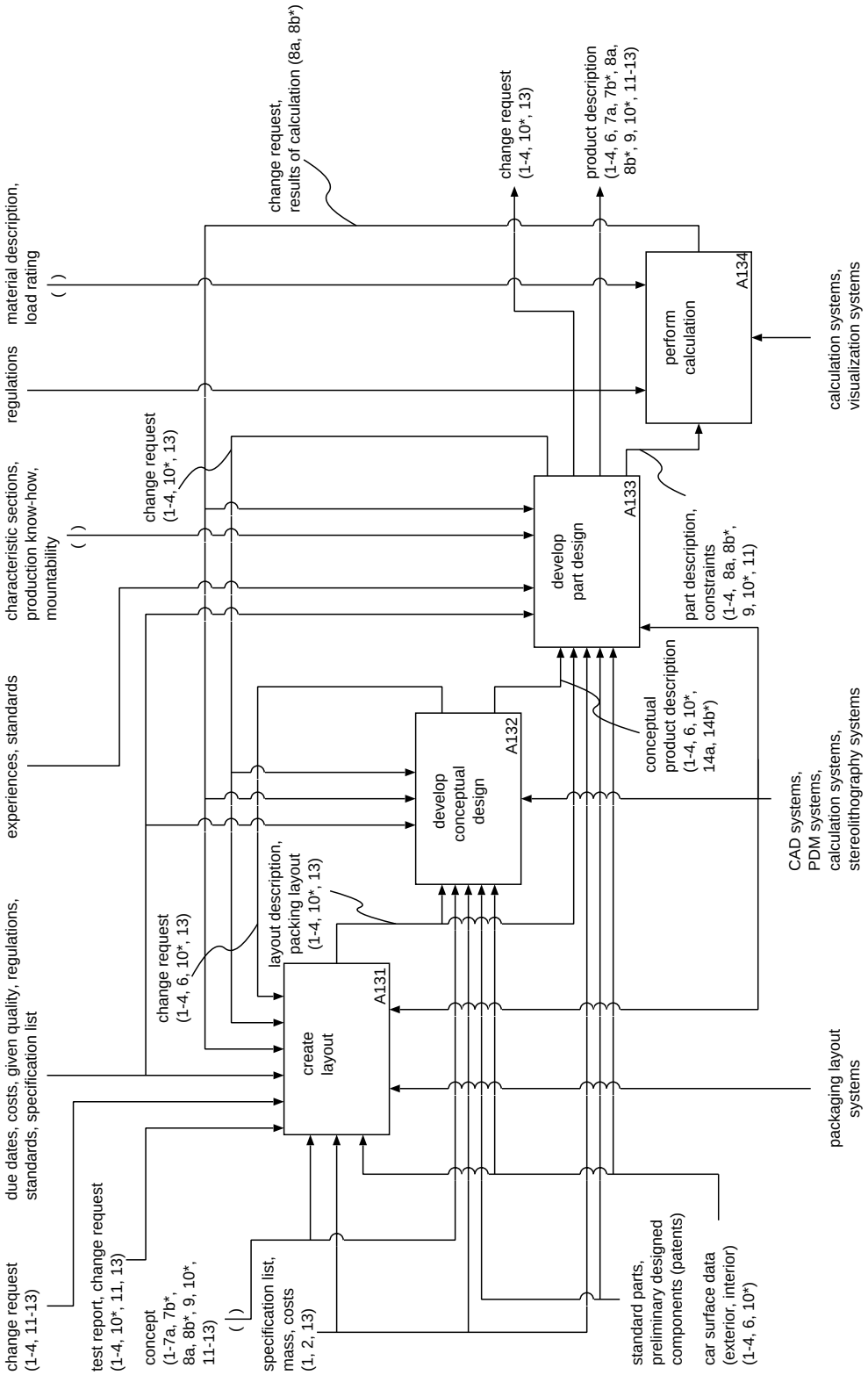


Figure F.5 – AAM diagram of node A13: Design product

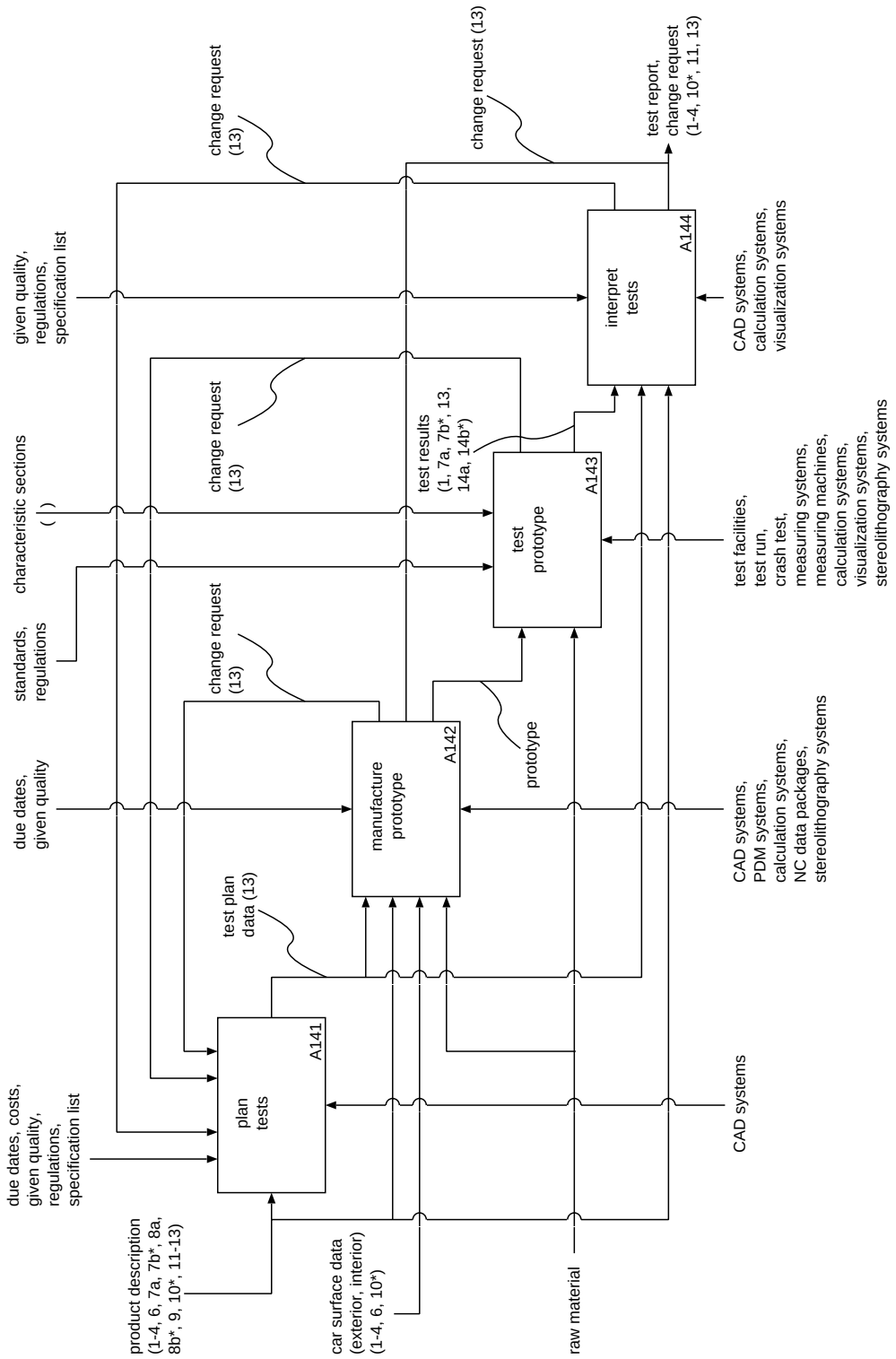


Figure F.6 – AAM diagram of node A14: Evaluate product

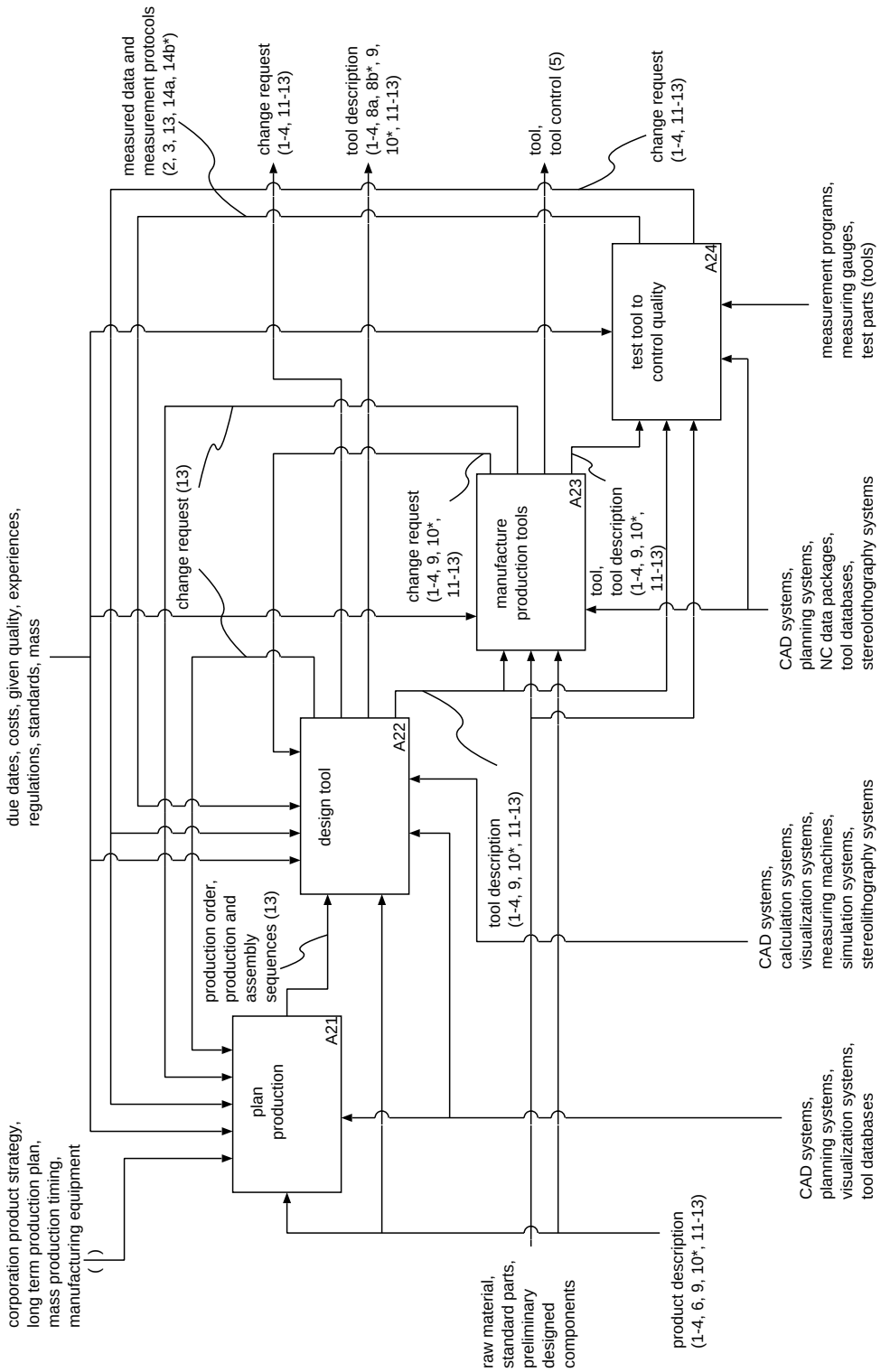


Figure F.7 – AAM diagram of node A2: Develop Tool

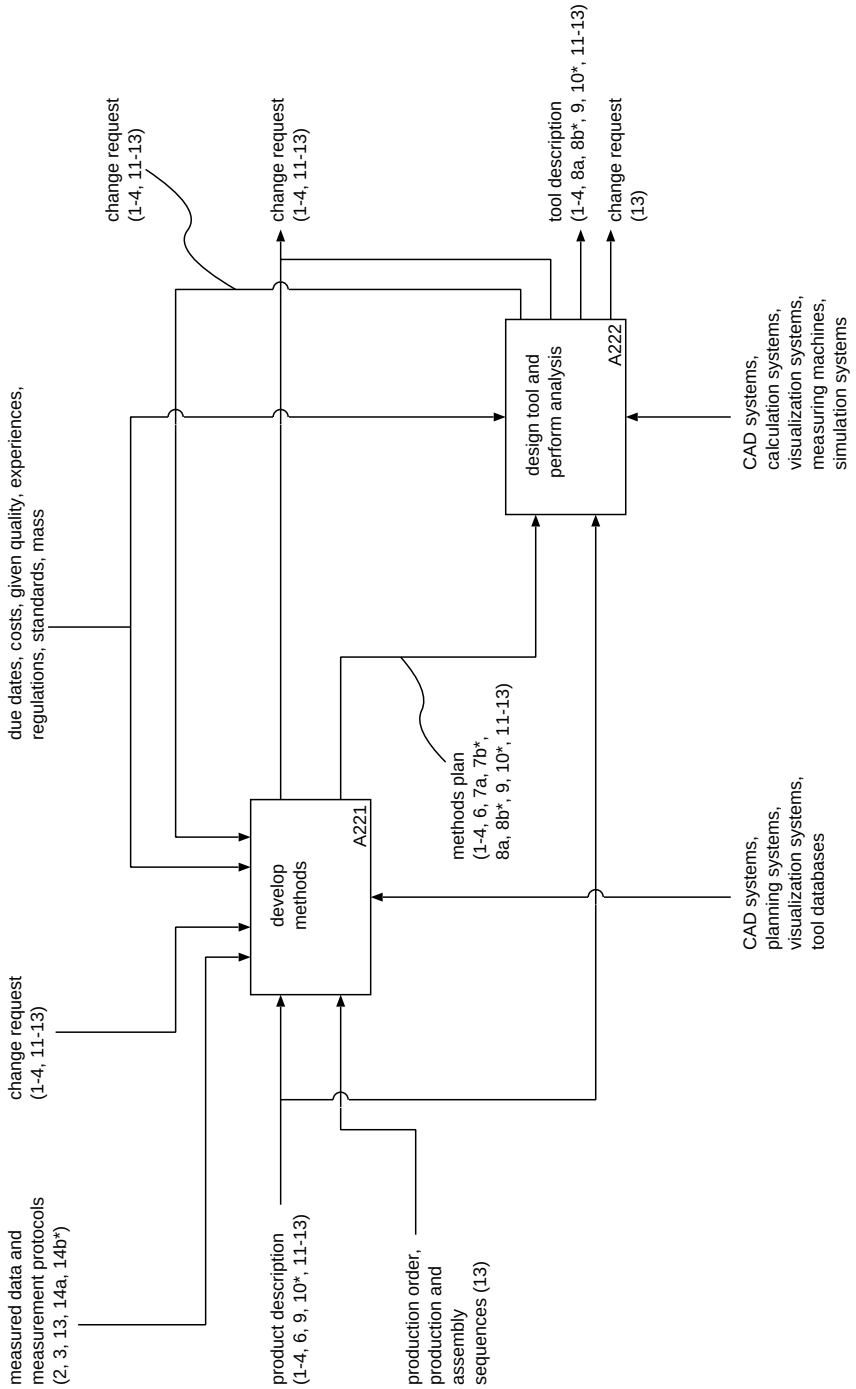


Figure F.8 – AAM diagram of node A22: Design tool

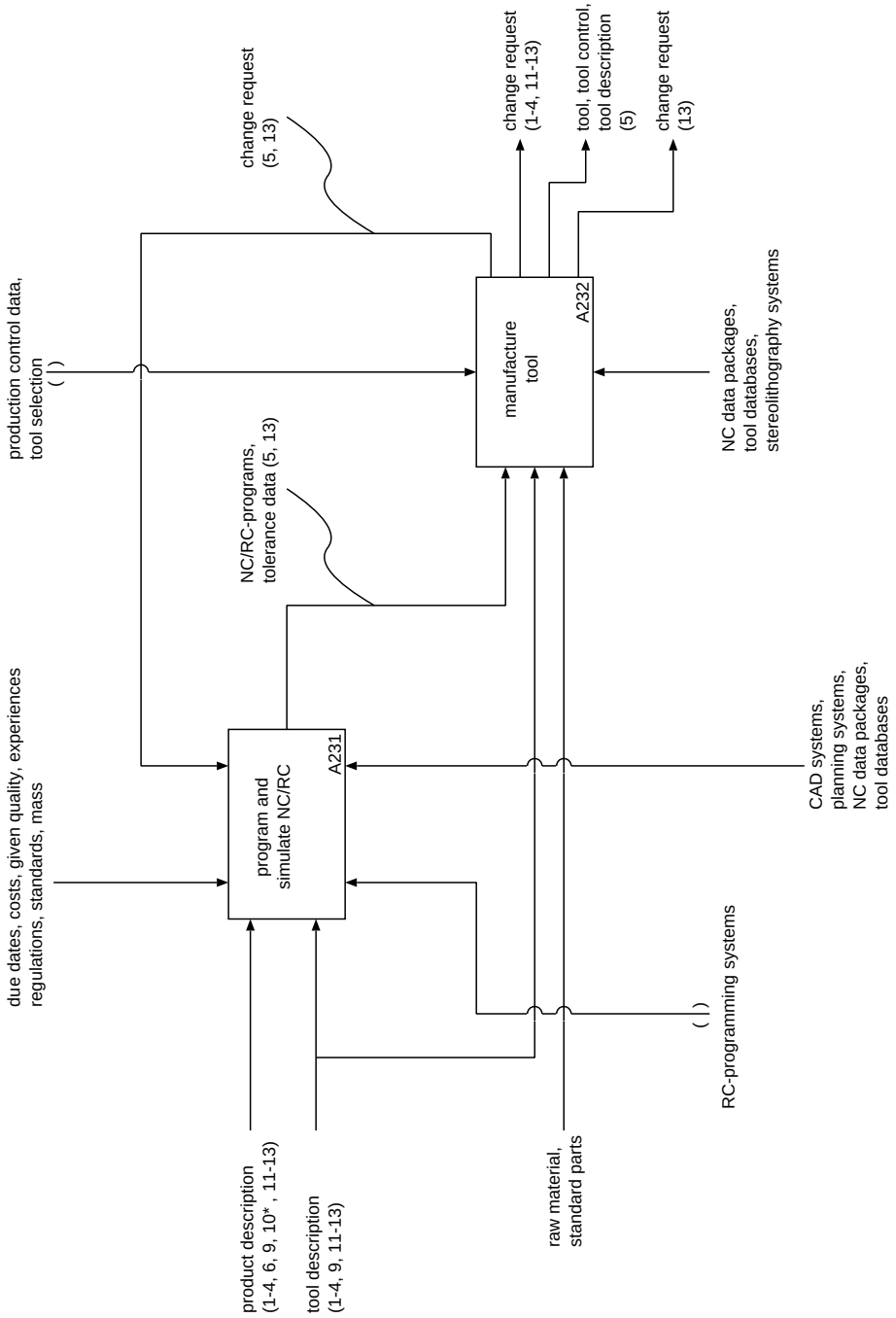


Figure F.9 – AAM diagram of node A23: Manufacture production tools

Annex G
(informative)
Application reference model

This annex provides the application reference model for this part of ISO 10303 and is given in figures G.1 through G.50. The application reference model is a graphical representation of the structure and constraints of the application objects specified in clause 4. The graphical form of the application reference model is presented in the EXPRESS-G format. The application reference model is independent from any implementation method.

NOTE 1 – Not all possible application assertions that are established through select types apply for this part of ISO 10303. See 4.3 for those application assertions that are in scope.

NOTE 2 – In EXPRESS-G it is not possible to express the inverse cardinalities for relationships where the relationship path includes select types. In those cases a footnote mark on the diagram refers to the end of this annex where additional information can be found.

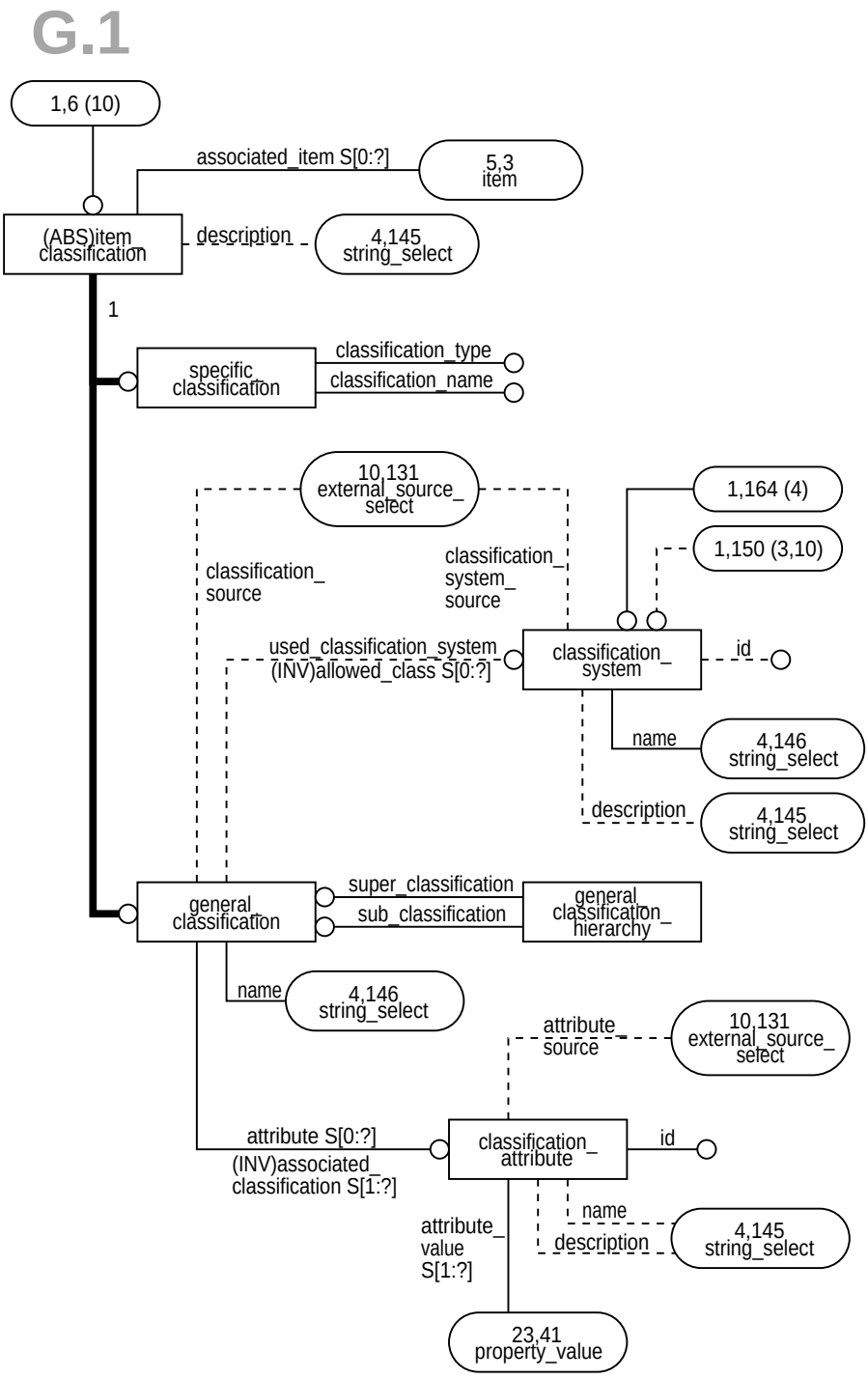


Figure G.1 – ARM diagram in EXPRESS_G: 1 of 50

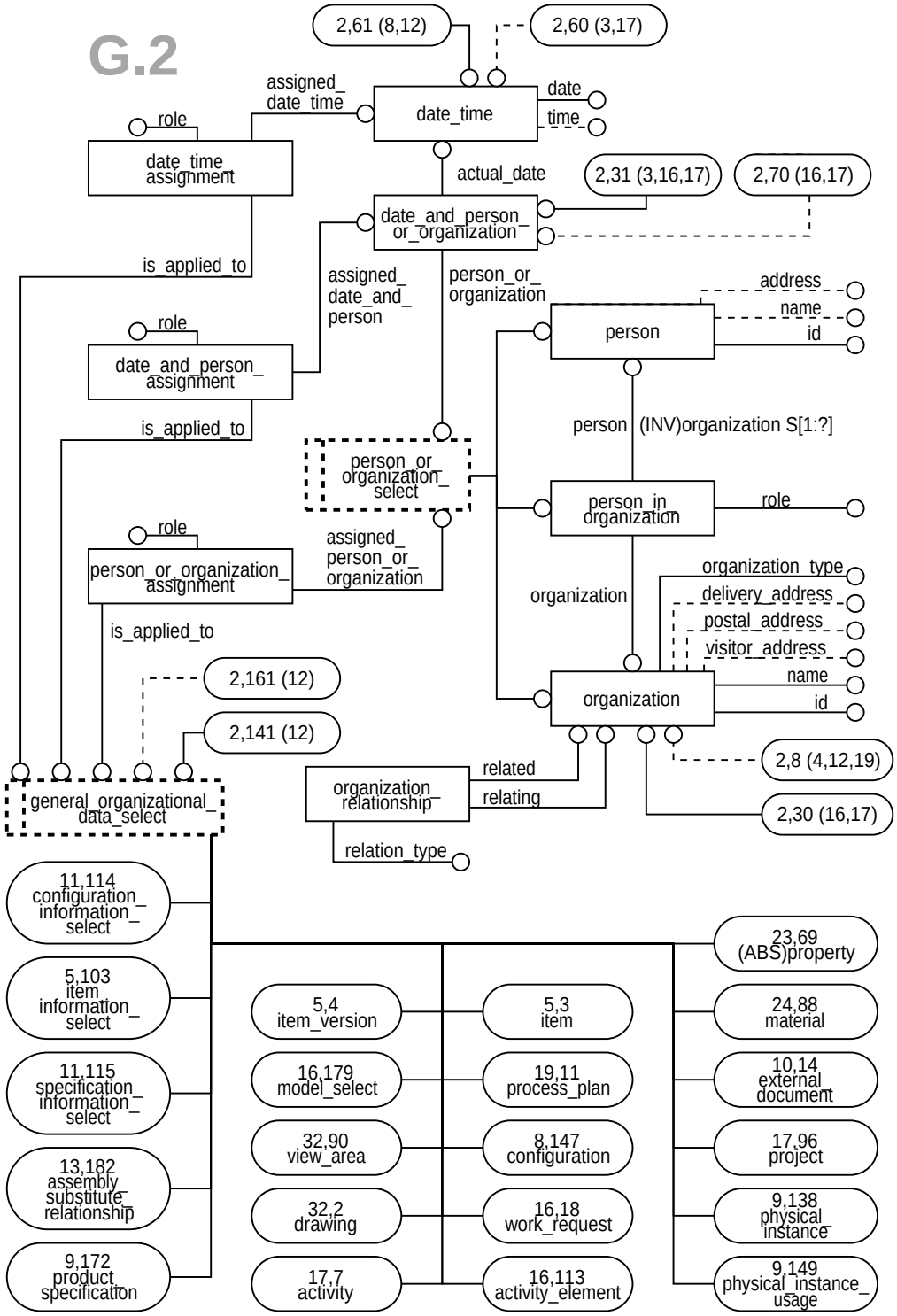


Figure G.2 – ARM diagram in EXPRESS_G: 2 of 50

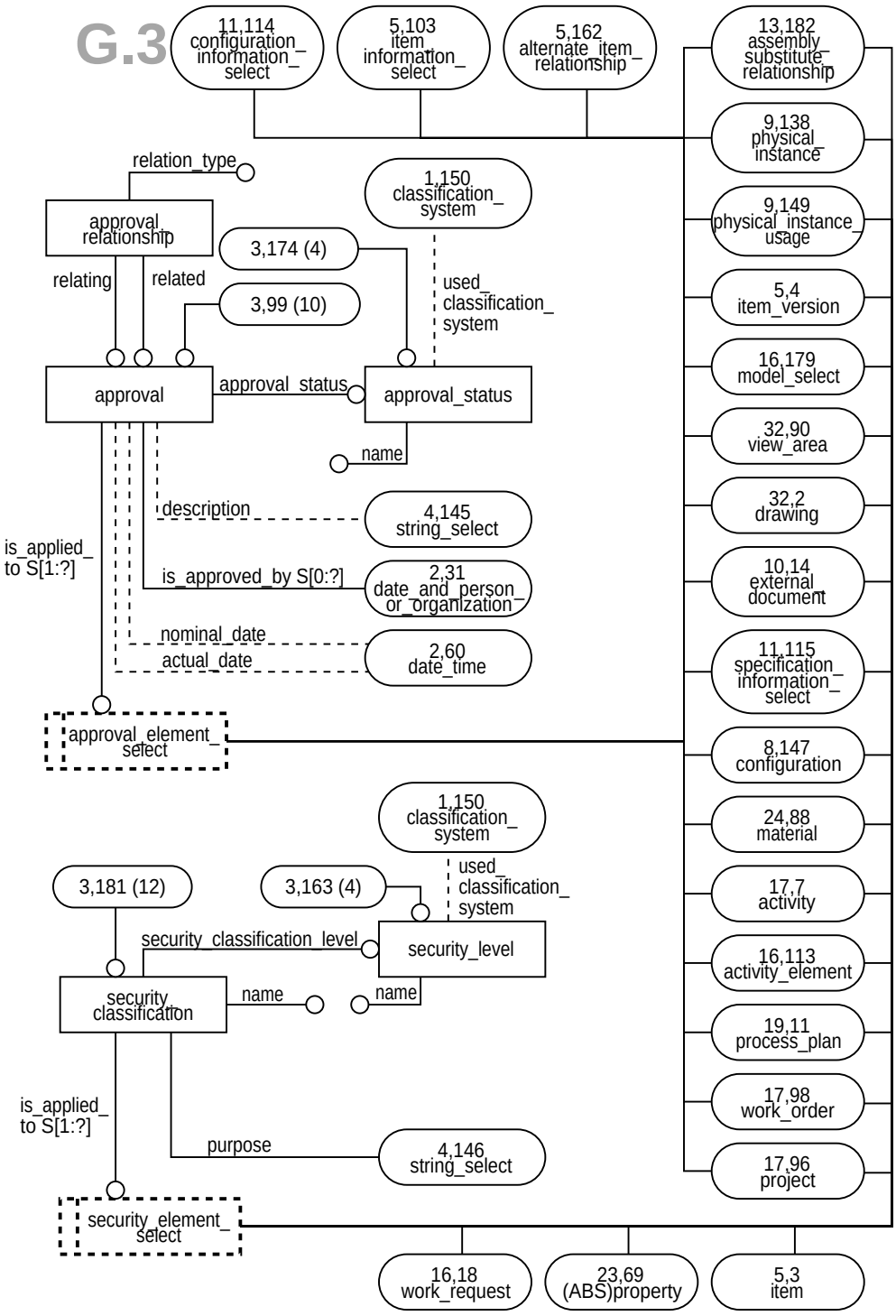


Figure G.3 – ARM diagram in EXPRESS_G: 3 of 50

G.4

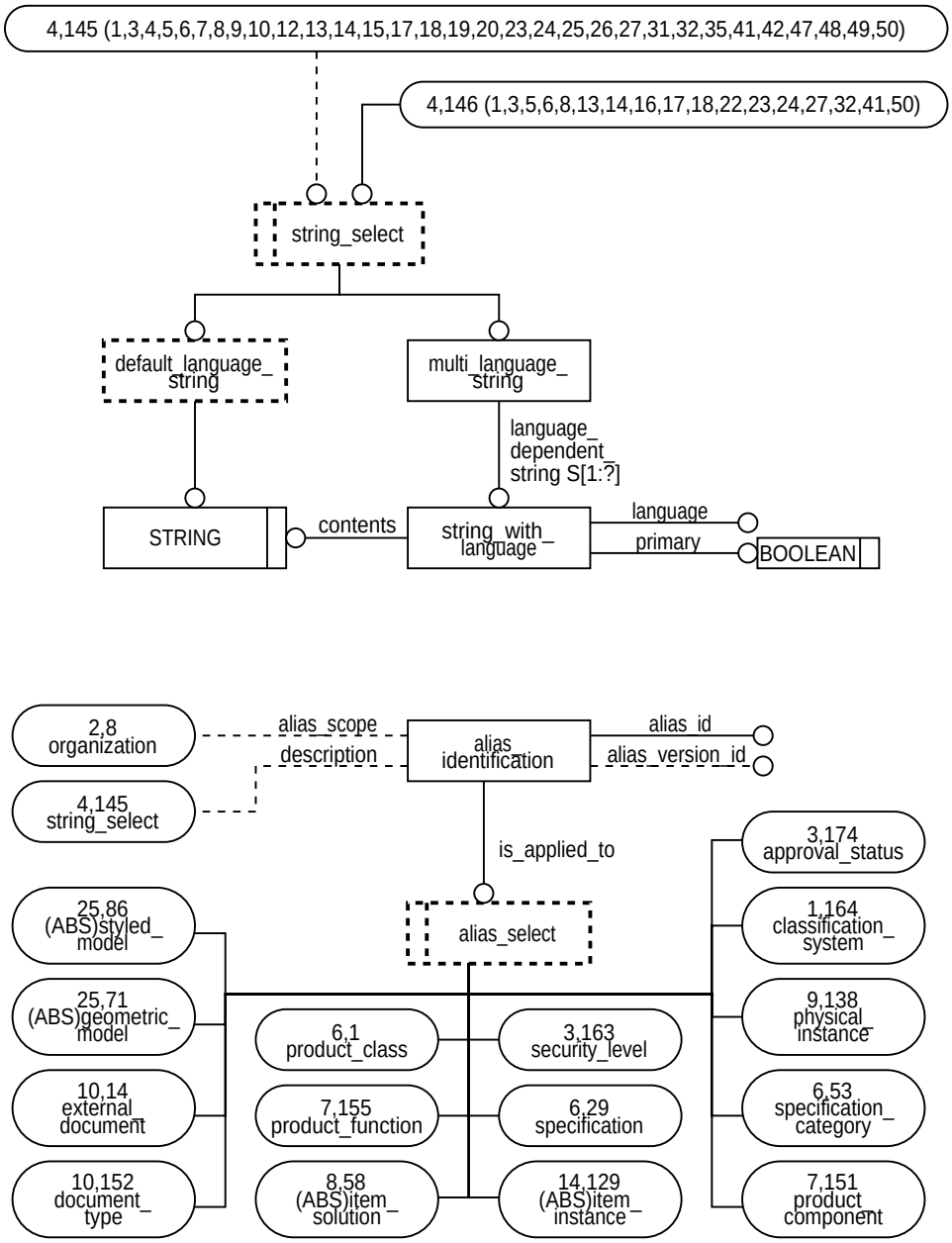


Figure G.4 – ARM diagram in EXPRESS_G: 4 of 50

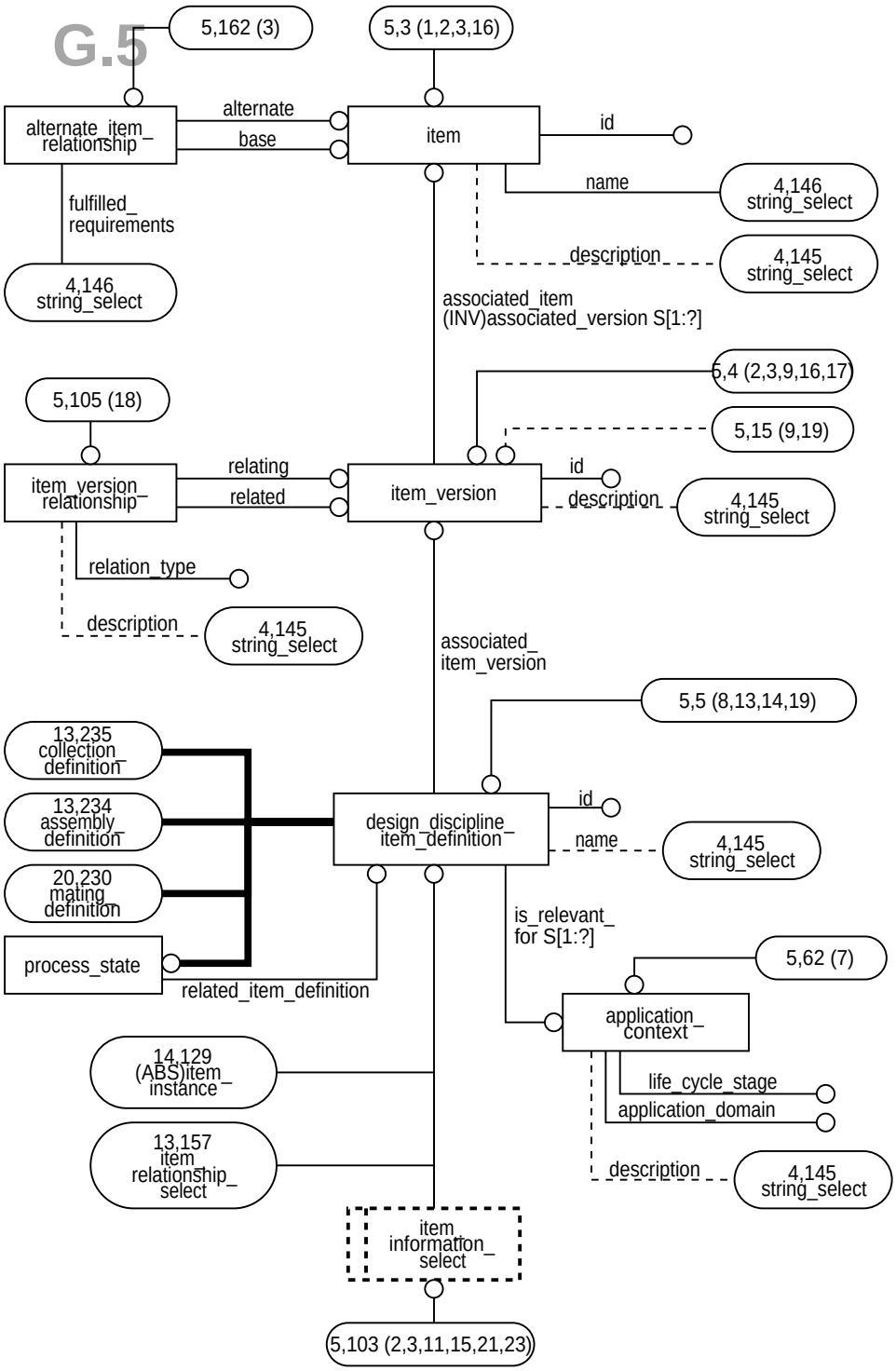


Figure G.5 – ARM diagram in EXPRESS_G: 5 of 50

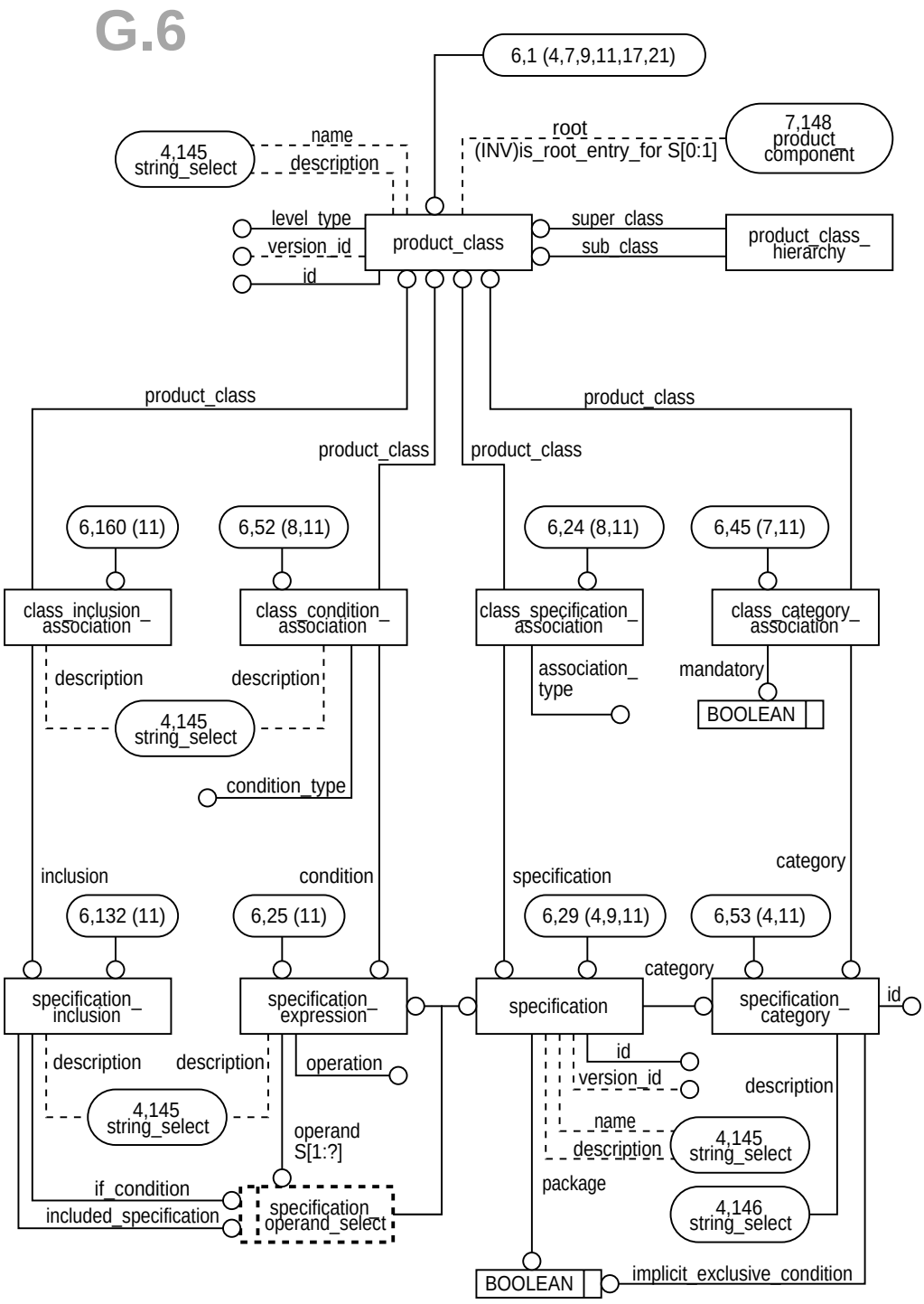


Figure G.6 – ARM diagram in EXPRESS_G: 6 of 50

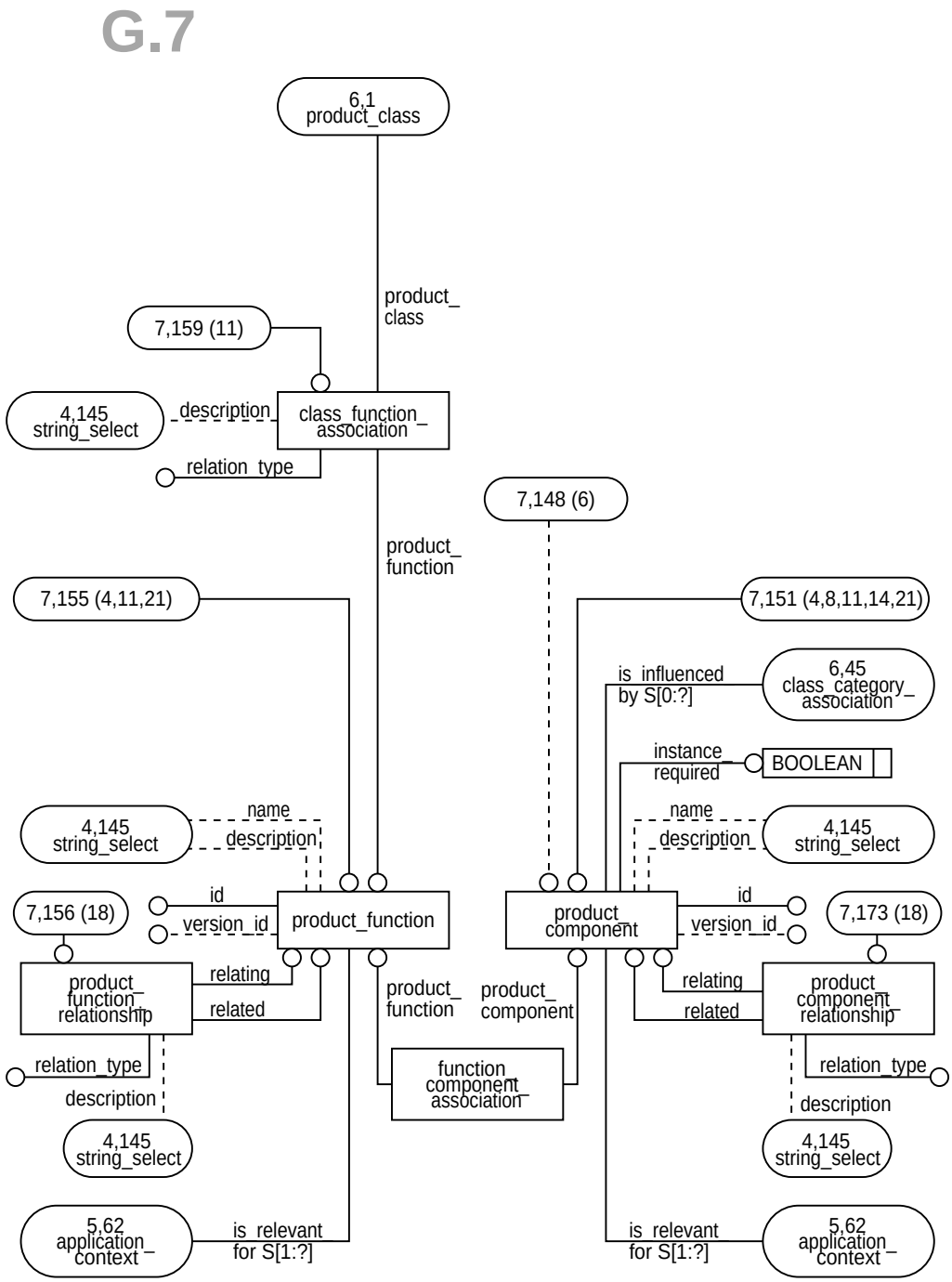


Figure G.7 – ARM diagram in EXPRESS_G: 7 of 50

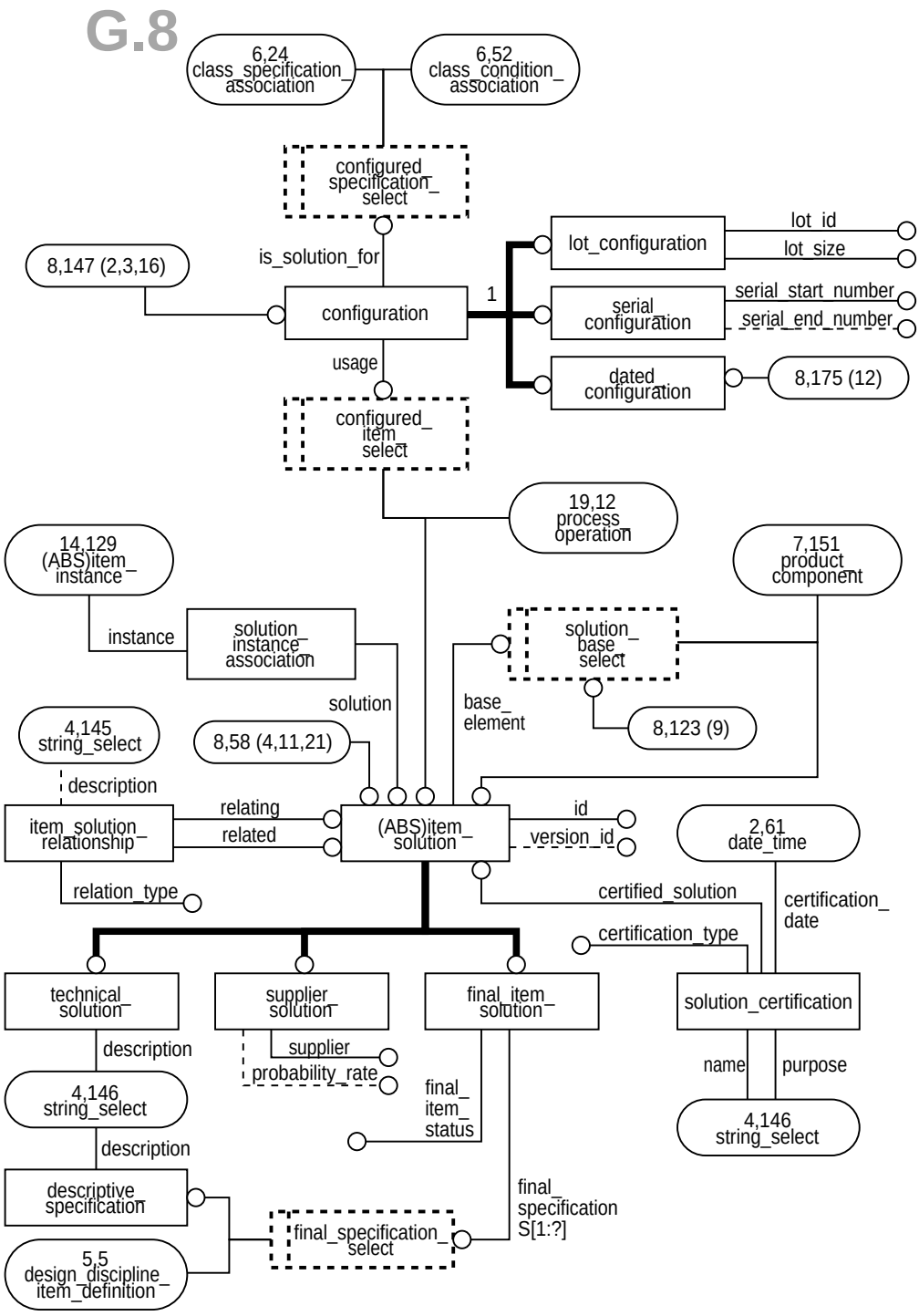


Figure G.8 – ARM diagram in EXPRESS_G: 8 of 50

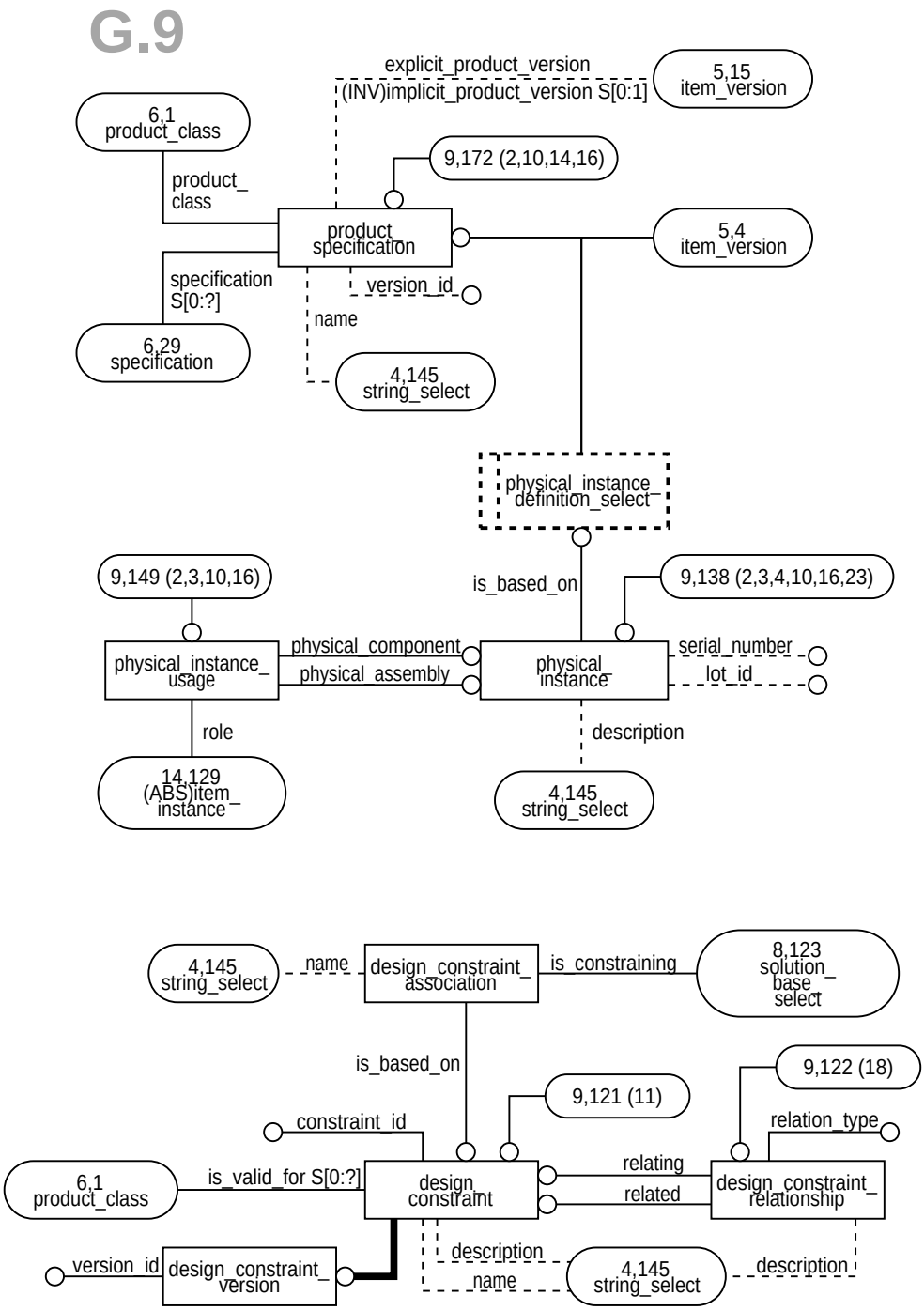


Figure G.9 – ARM diagram in EXPRESS_G: 9 of 50

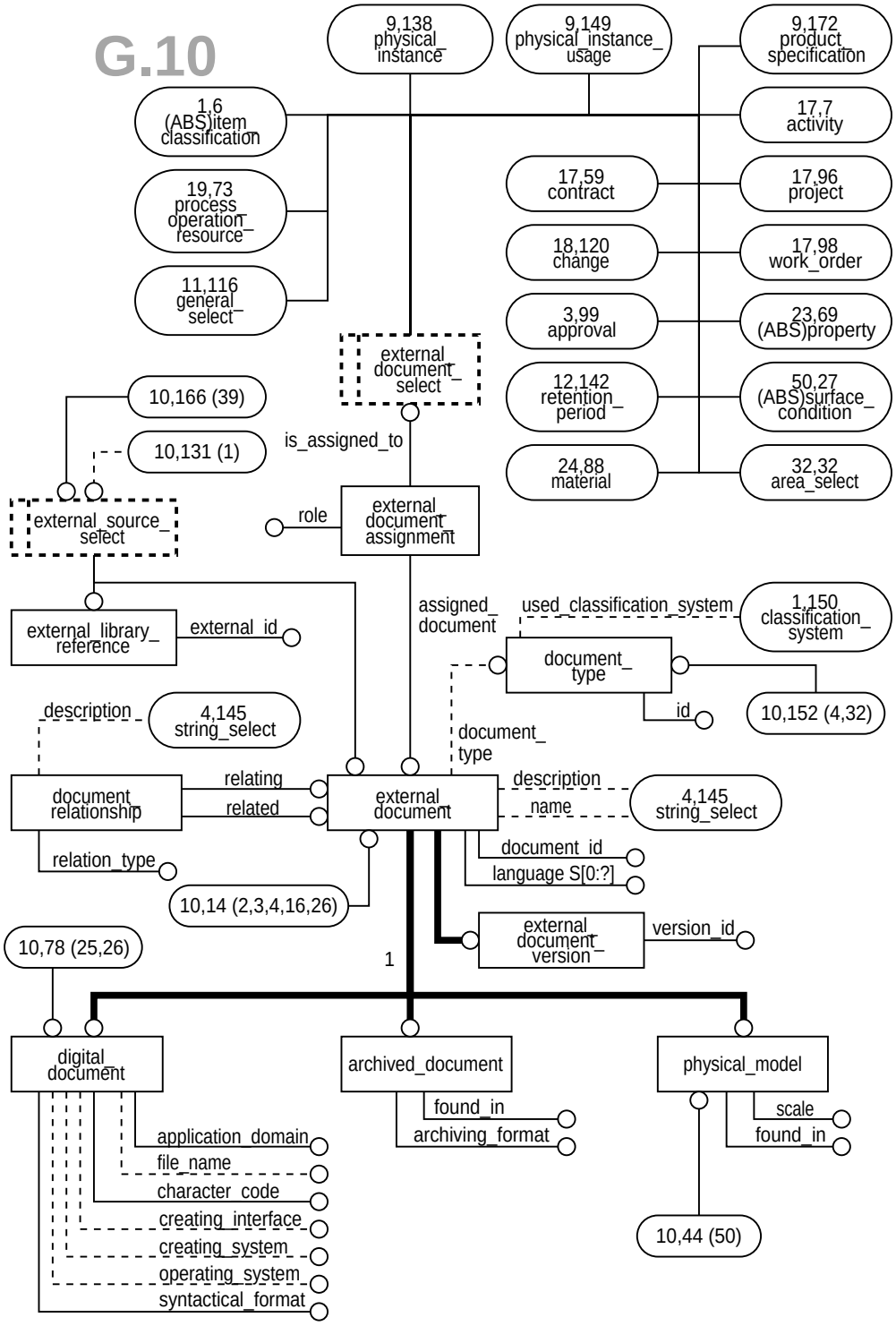


Figure G.10 – ARM diagram in EXPRESS_G: 10 of 50

G.11

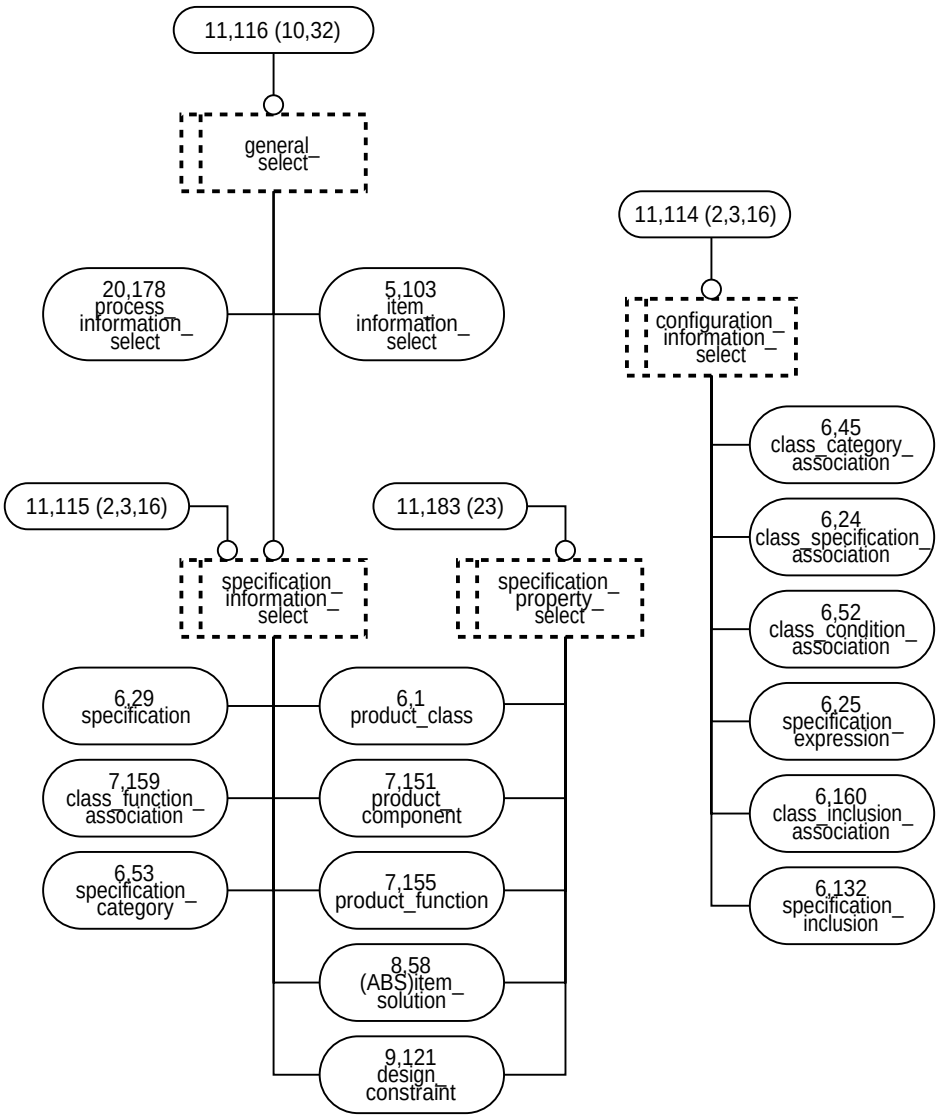


Figure G.11 – ARM diagram in EXPRESS_G: 11 of 50

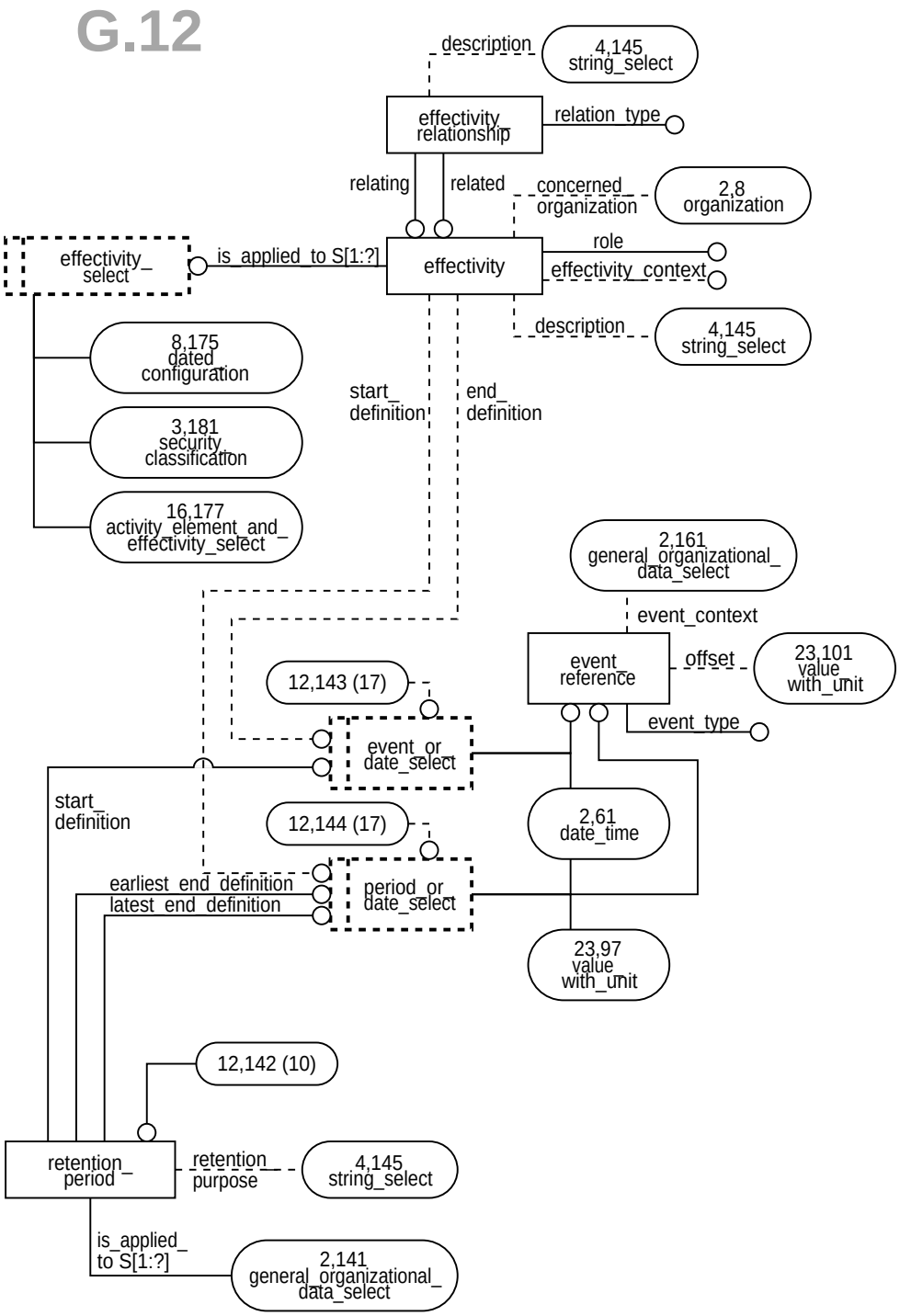


Figure G.12 – ARM diagram in EXPRESS_G: 12 of 50

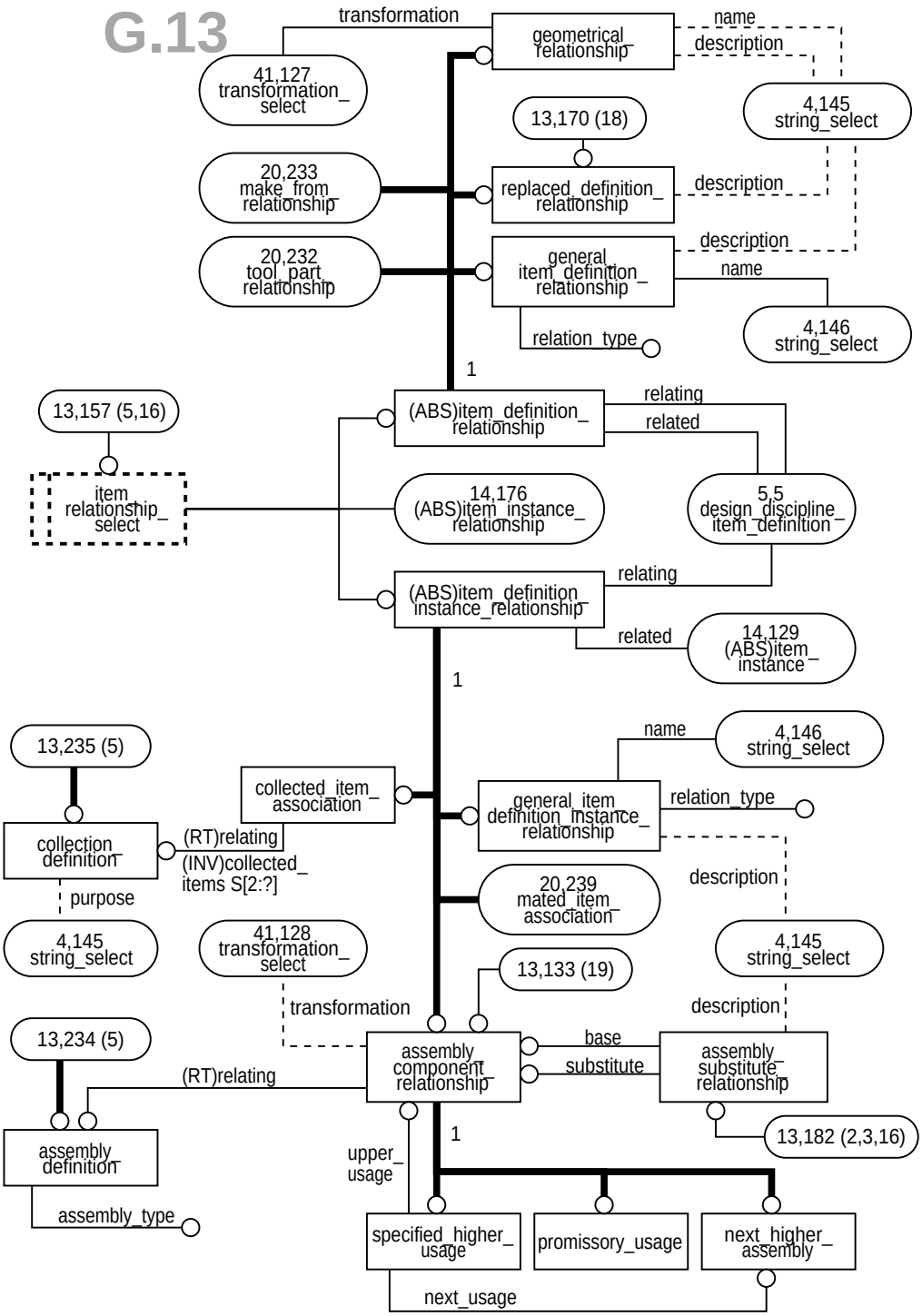


Figure G.13 – ARM diagram in EXPRESS_G: 13 of 50

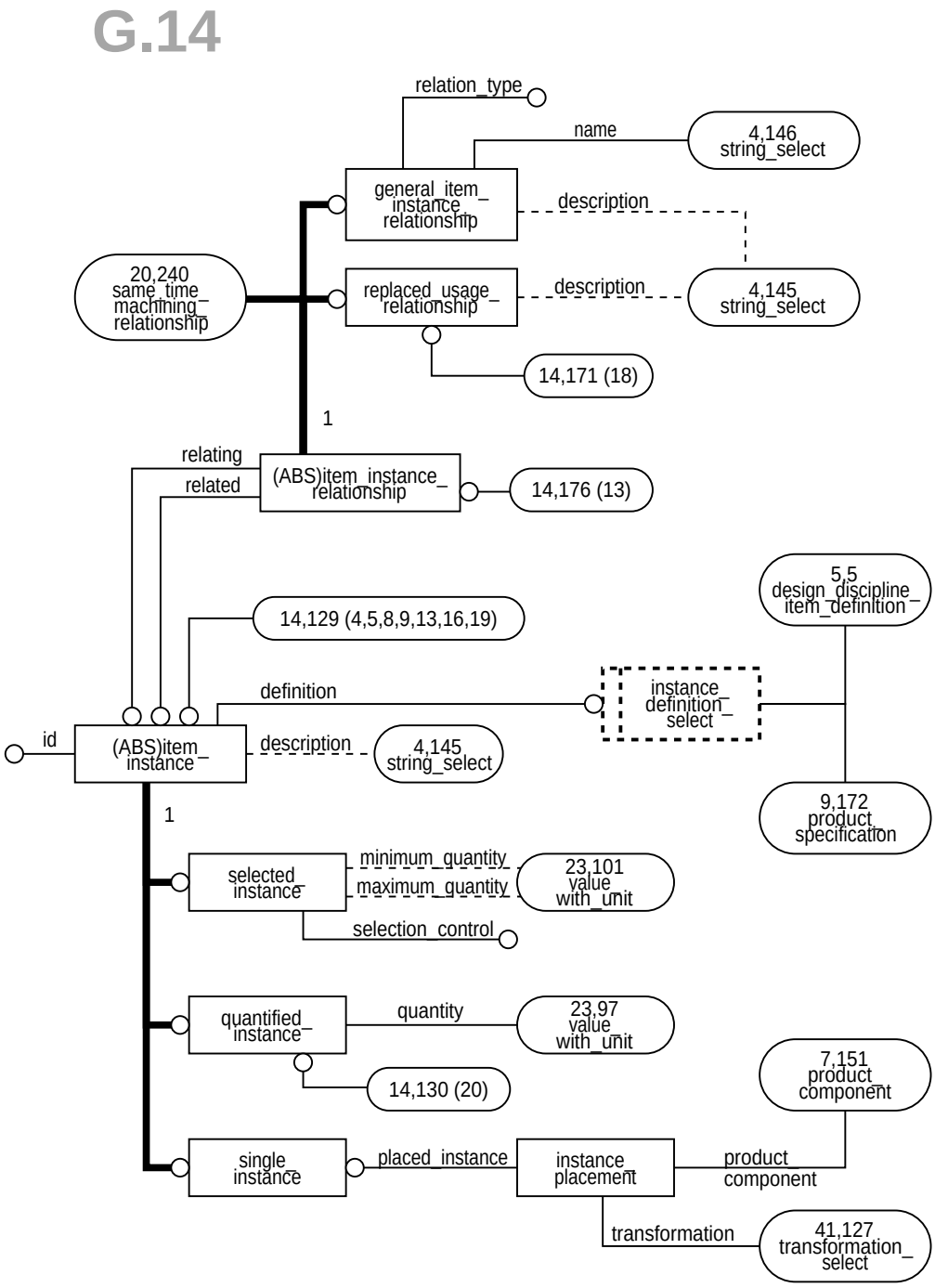


Figure G.14 – ARM diagram in EXPRESS_G: 14 of 50

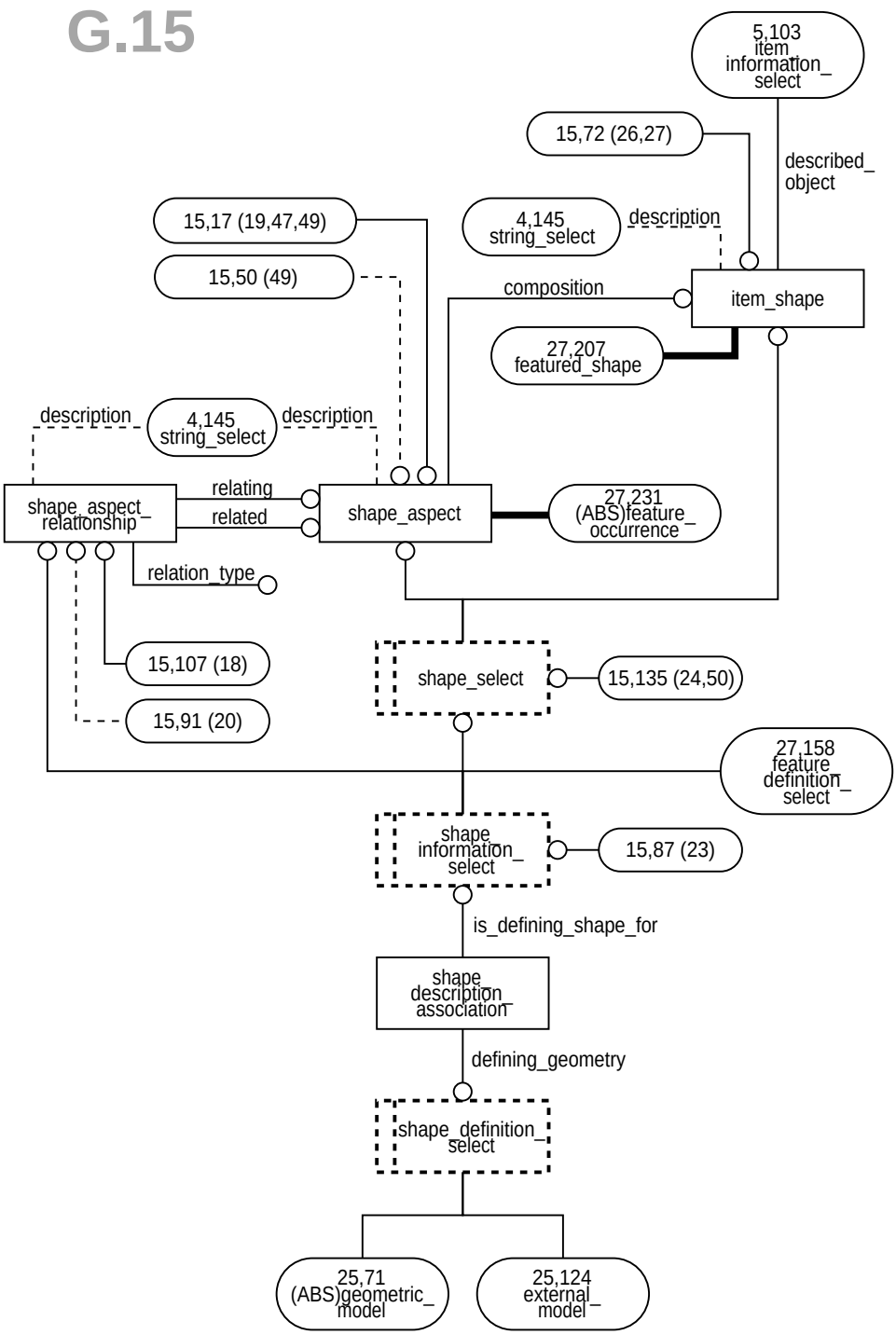


Figure G.15 – ARM diagram in EXPRESS_G: 15 of 50

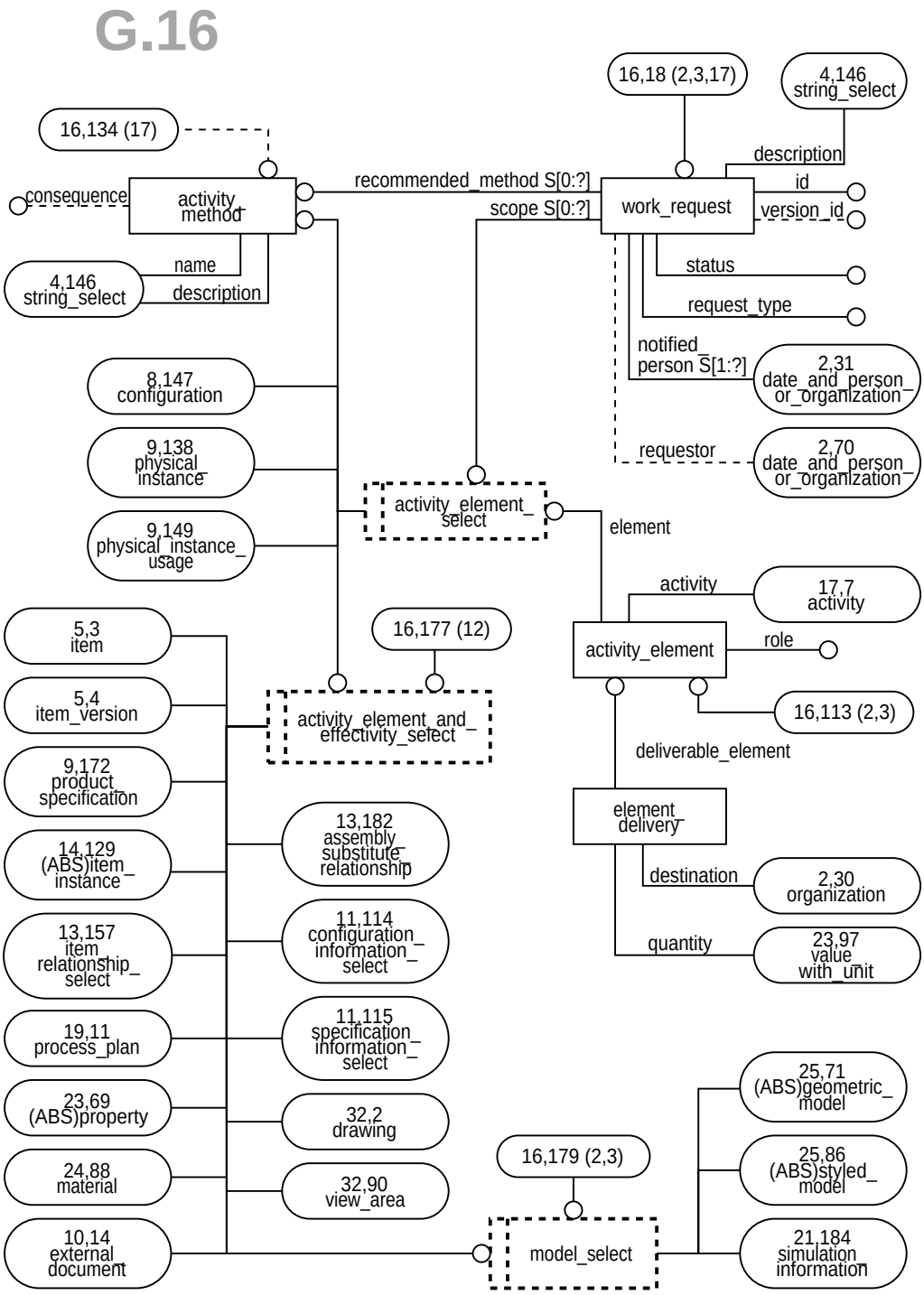


Figure G.16 – ARM diagram in EXPRESS_G: 16 of 50

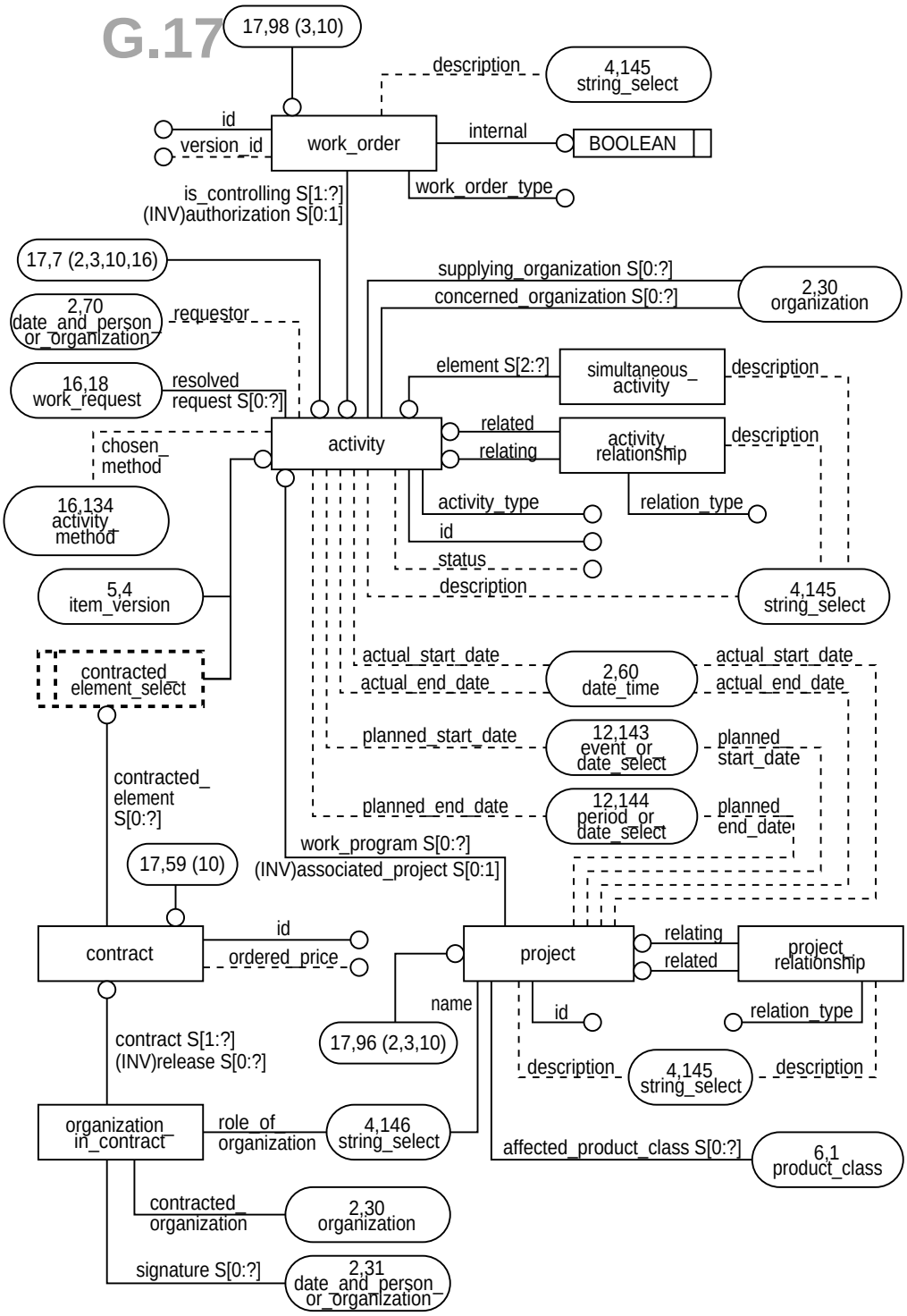


Figure G.17 – ARM diagram in EXPRESS_G: 17 of 50

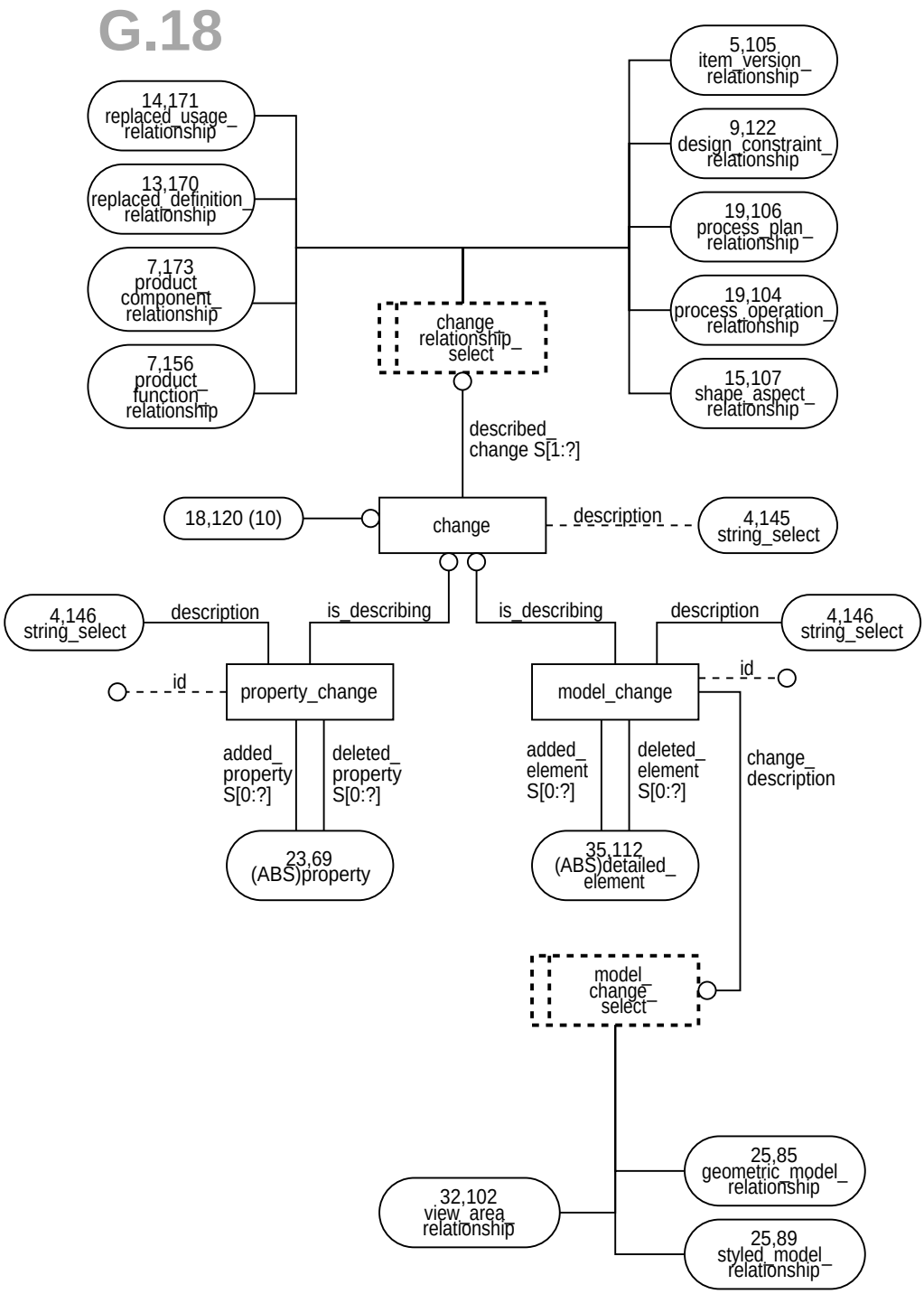


Figure G.18 – ARM diagram in EXPRESS_G: 18 of 50

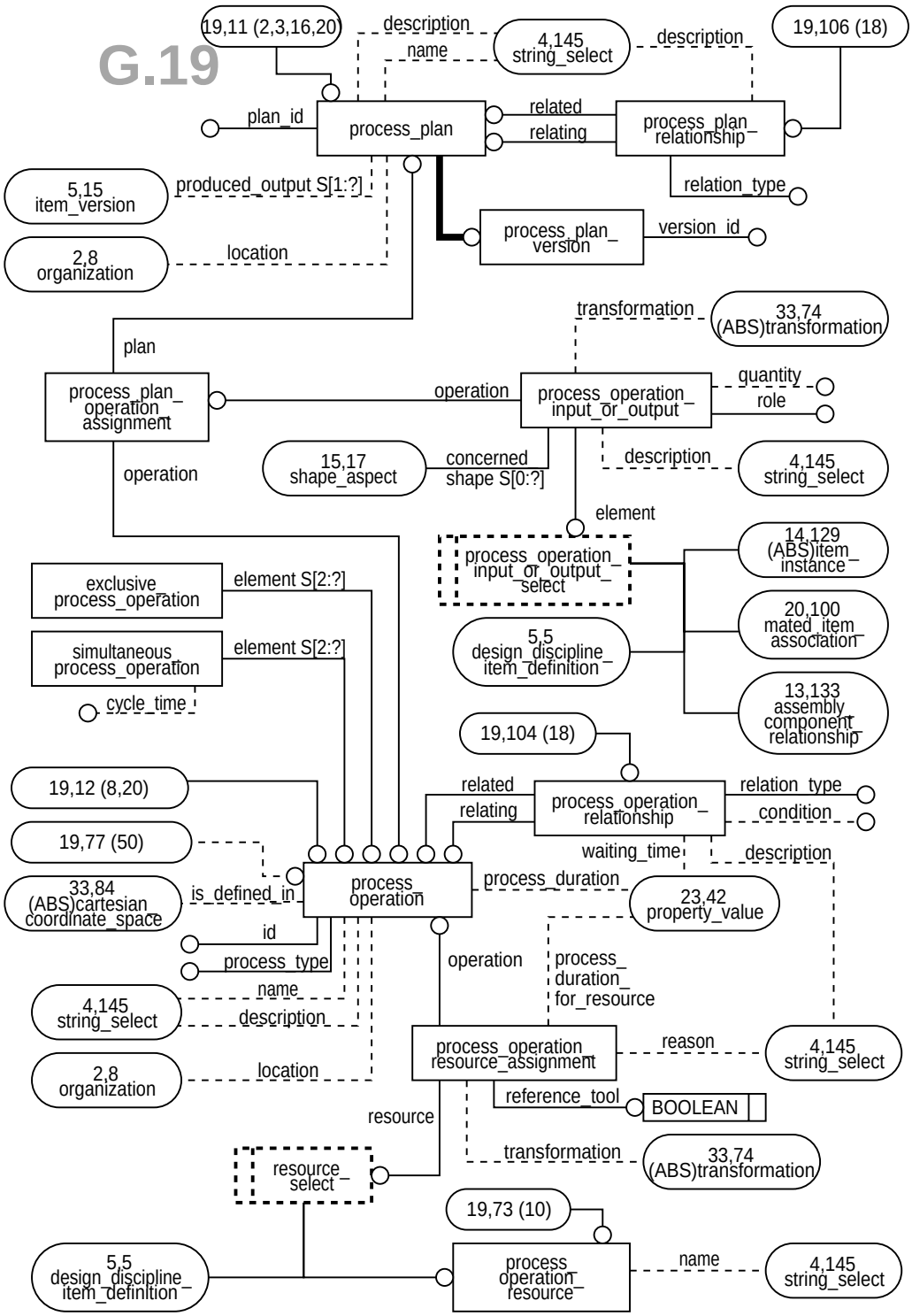


Figure G.19 – ARM diagram in EXPRESS_G: 19 of 50

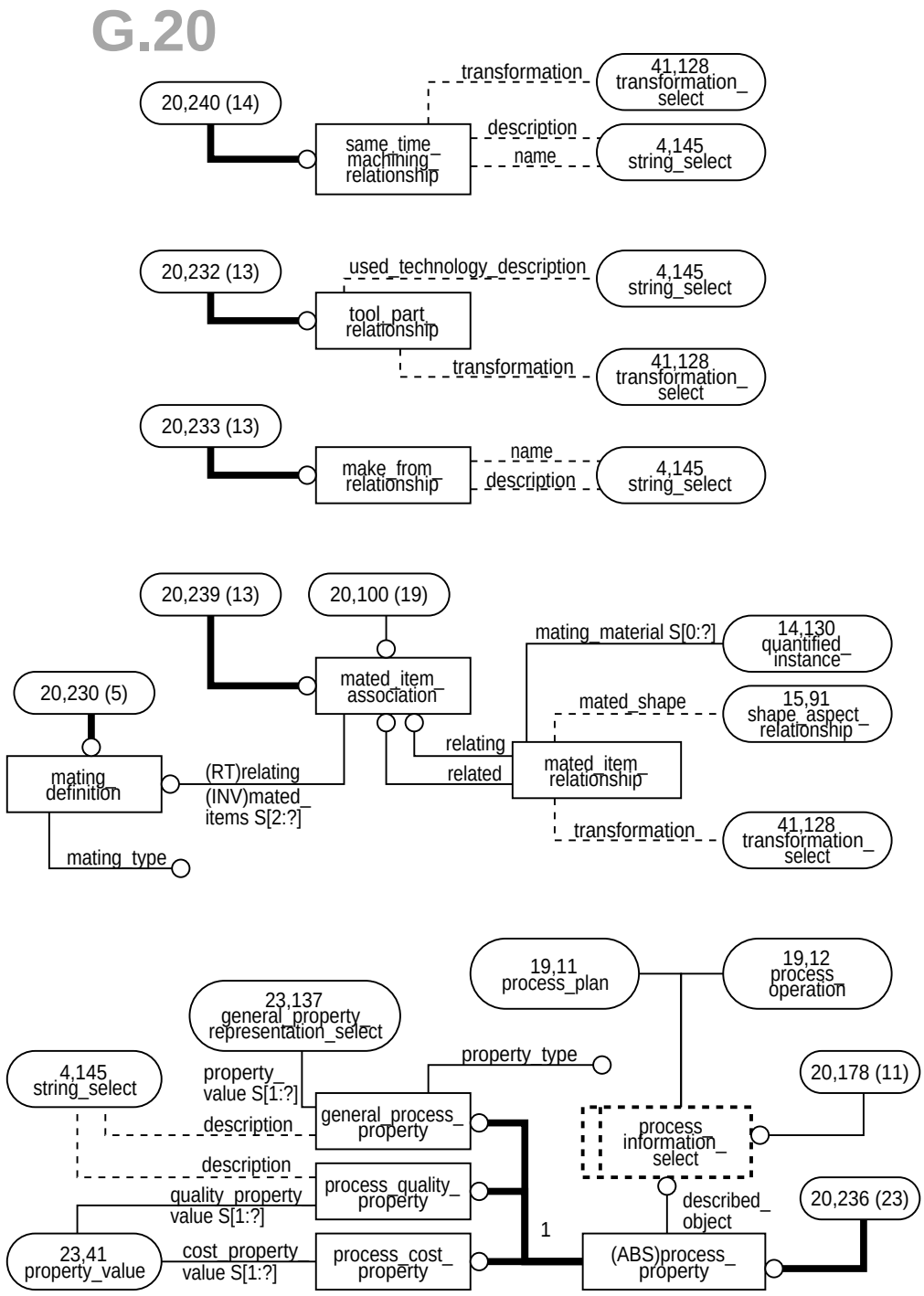


Figure G.20 – ARM diagram in EXPRESS_G: 20 of 50

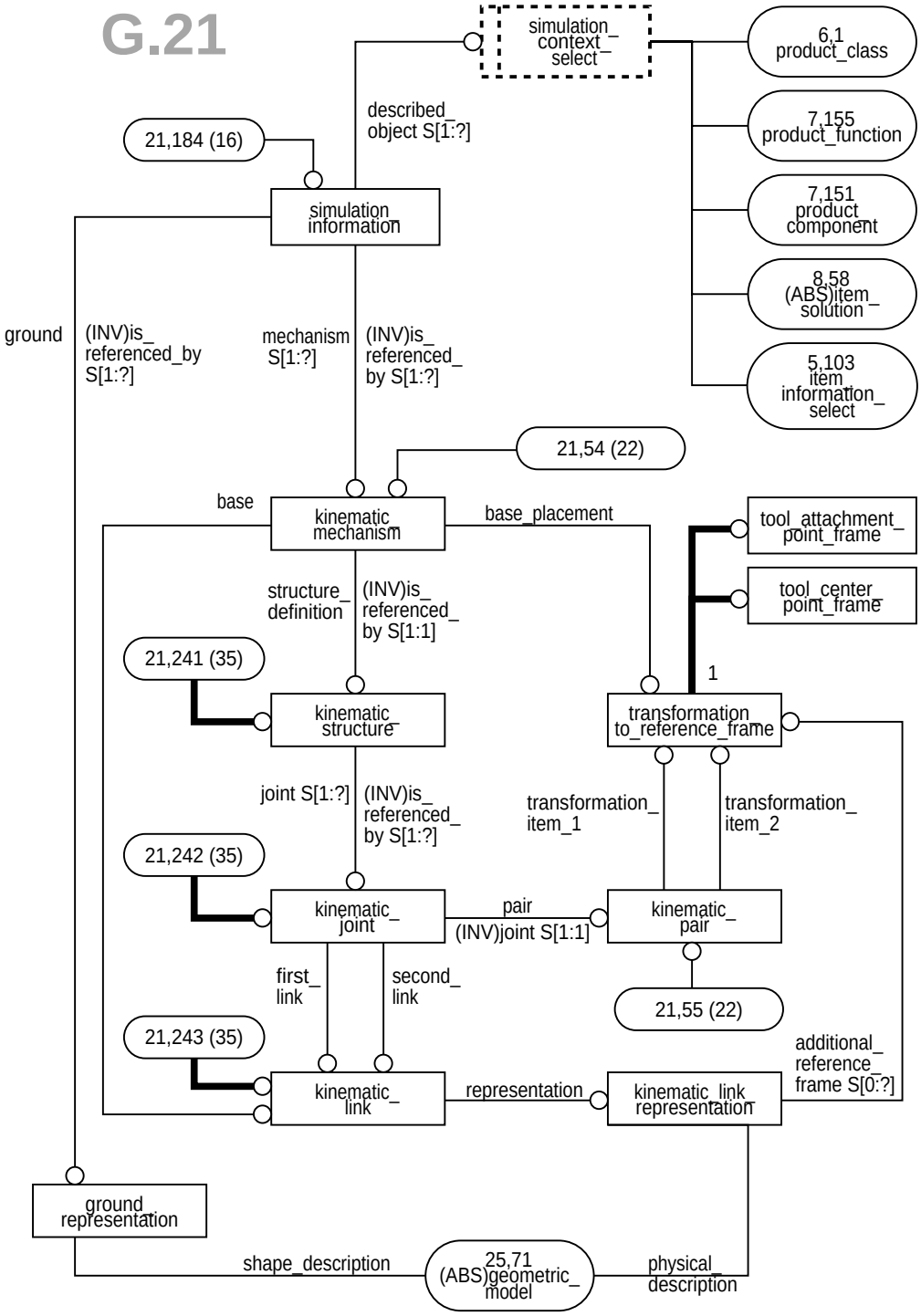


Figure G.21 – ARM diagram in EXPRESS_G: 21 of 50

G.22

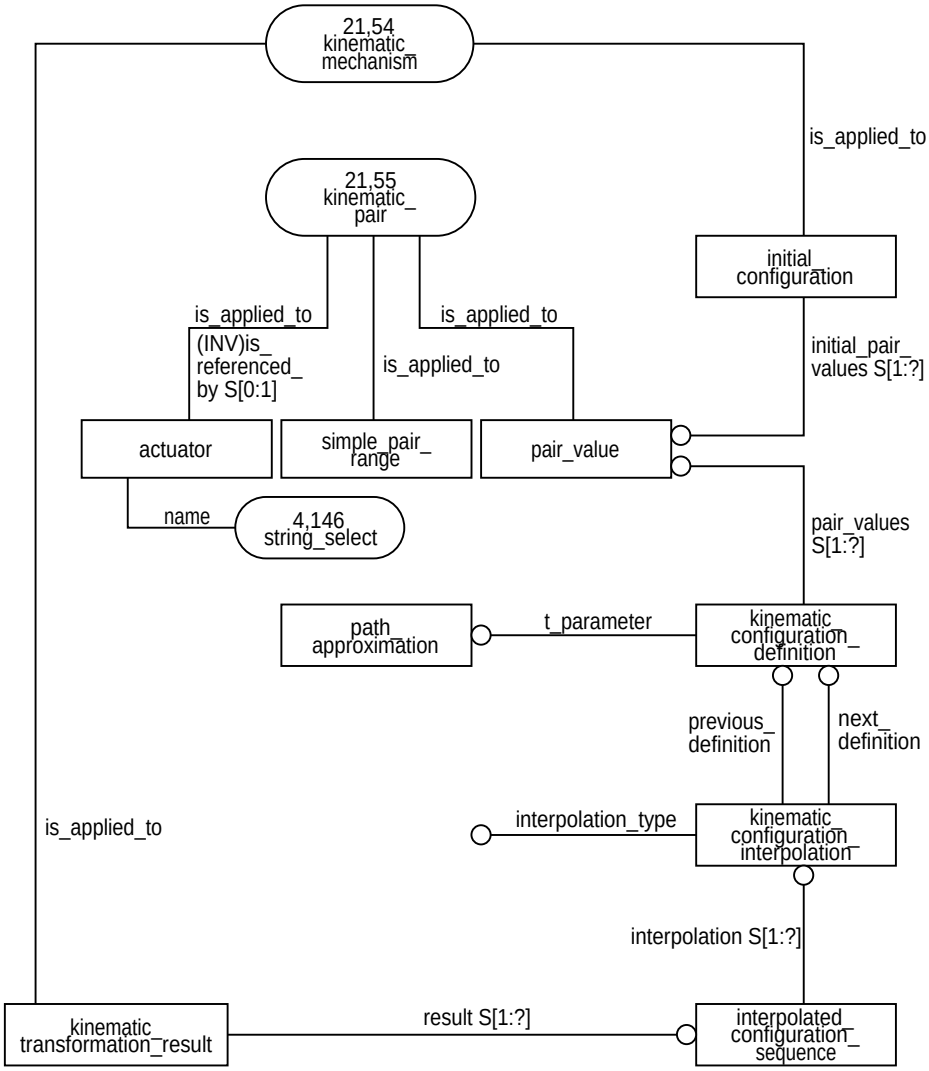


Figure G.22 – ARM diagram in EXPRESS_G: 22 of 50

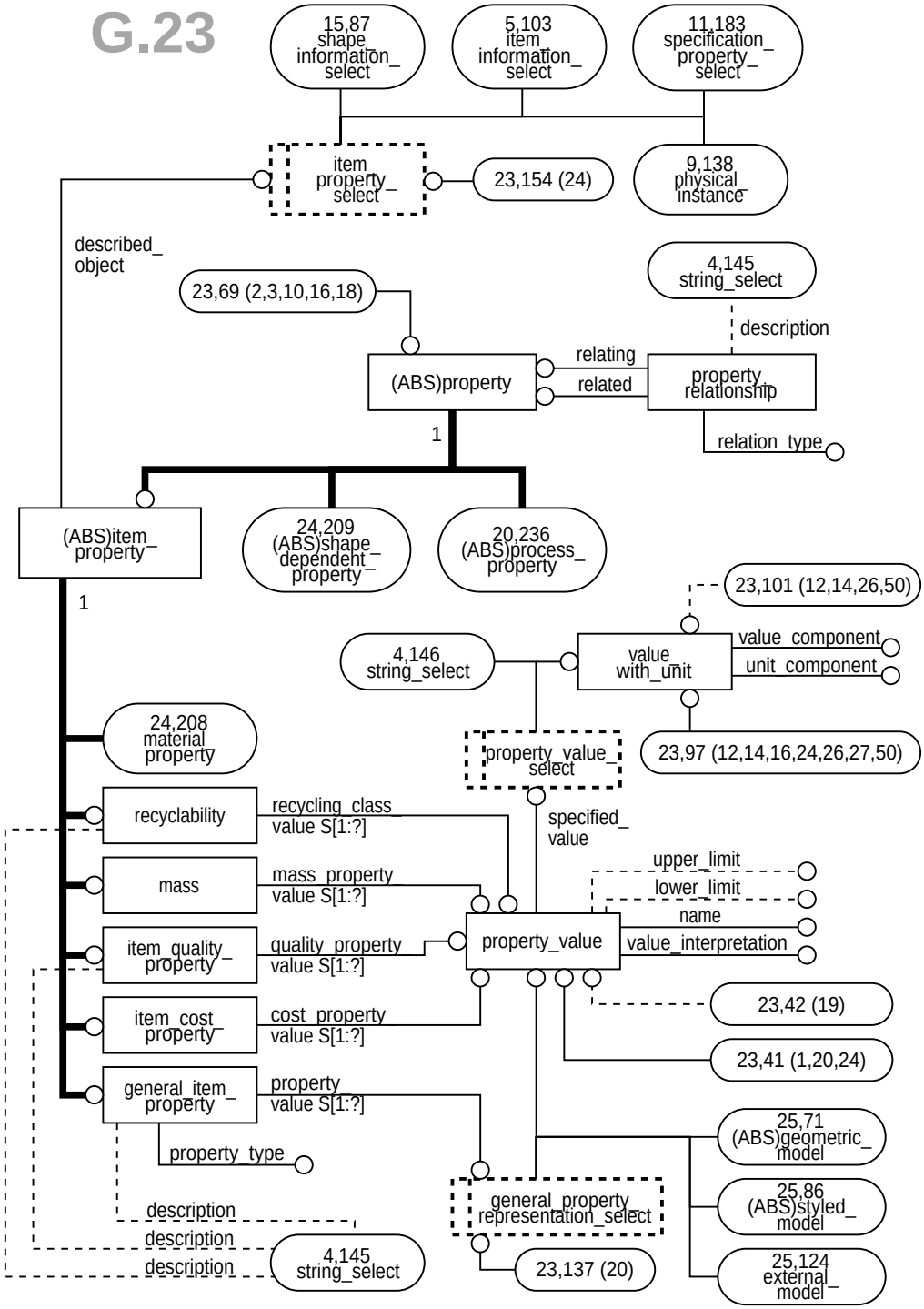


Figure G.23 – ARM diagram in EXPRESS_G: 23 of 50

G.24

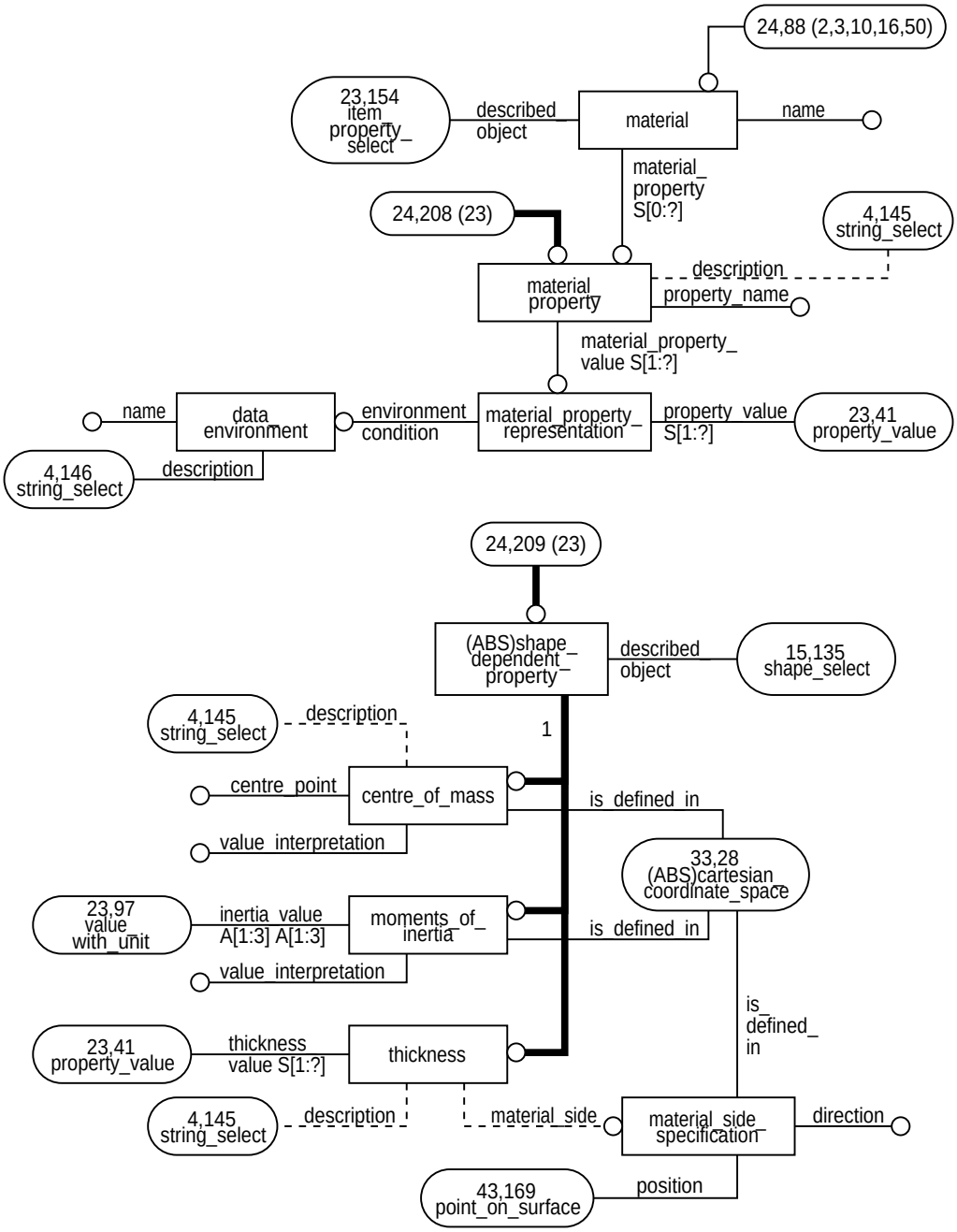


Figure G.24 – ARM diagram in EXPRESS_G: 24 of 50

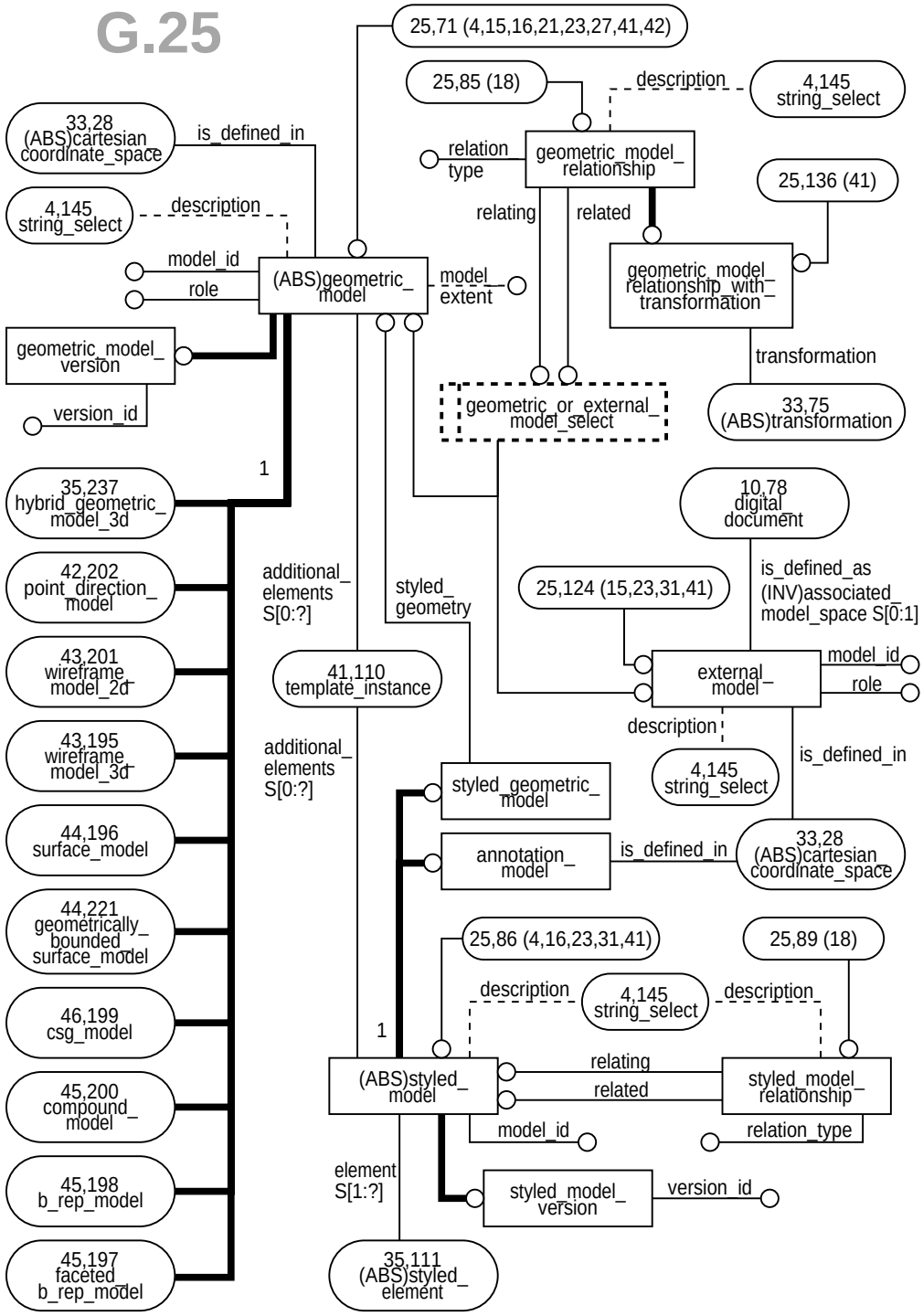


Figure G.25 – ARM diagram in EXPRESS_G: 25 of 50

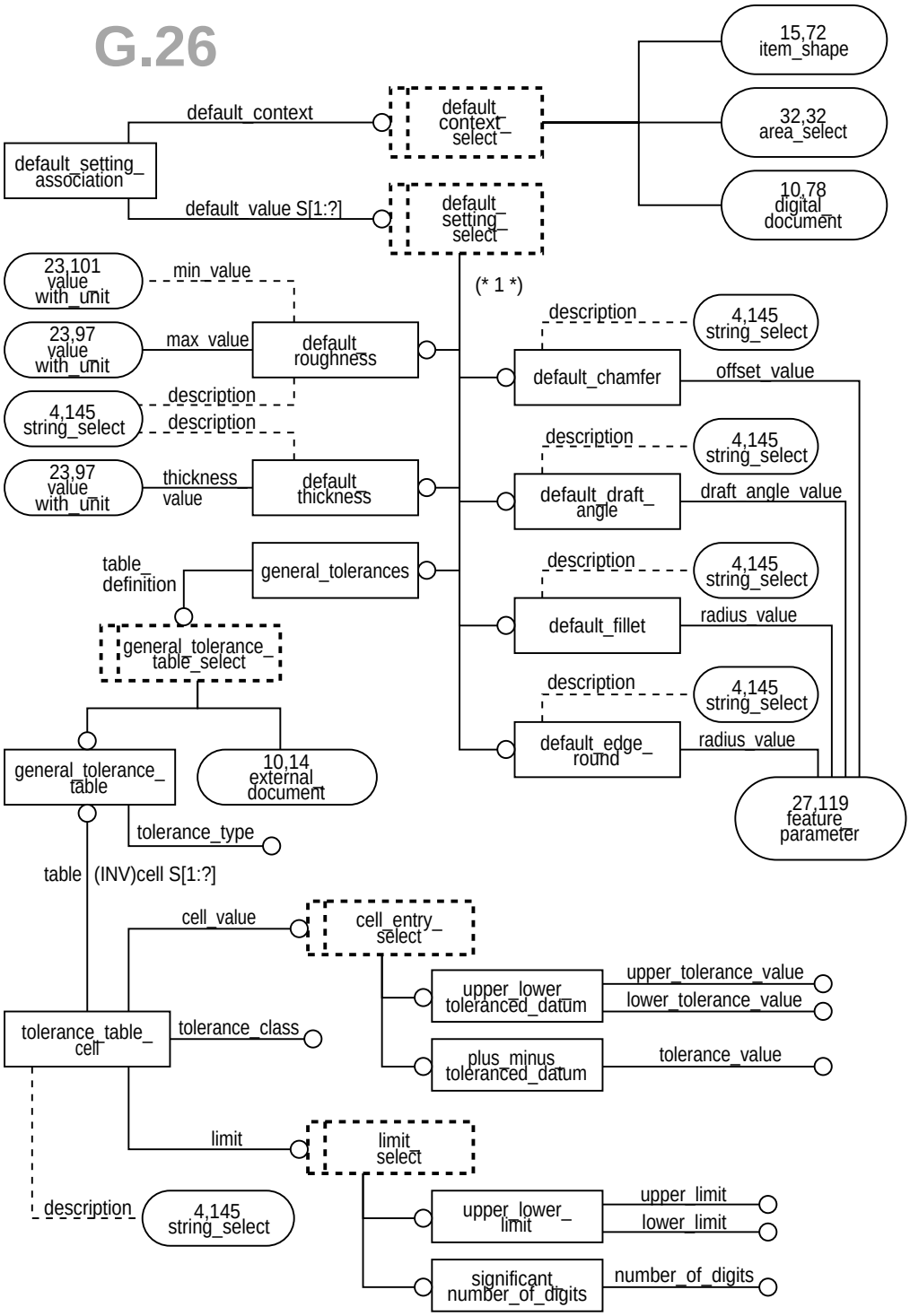


Figure G.26 – ARM diagram in EXPRESS_G: 26 of 50

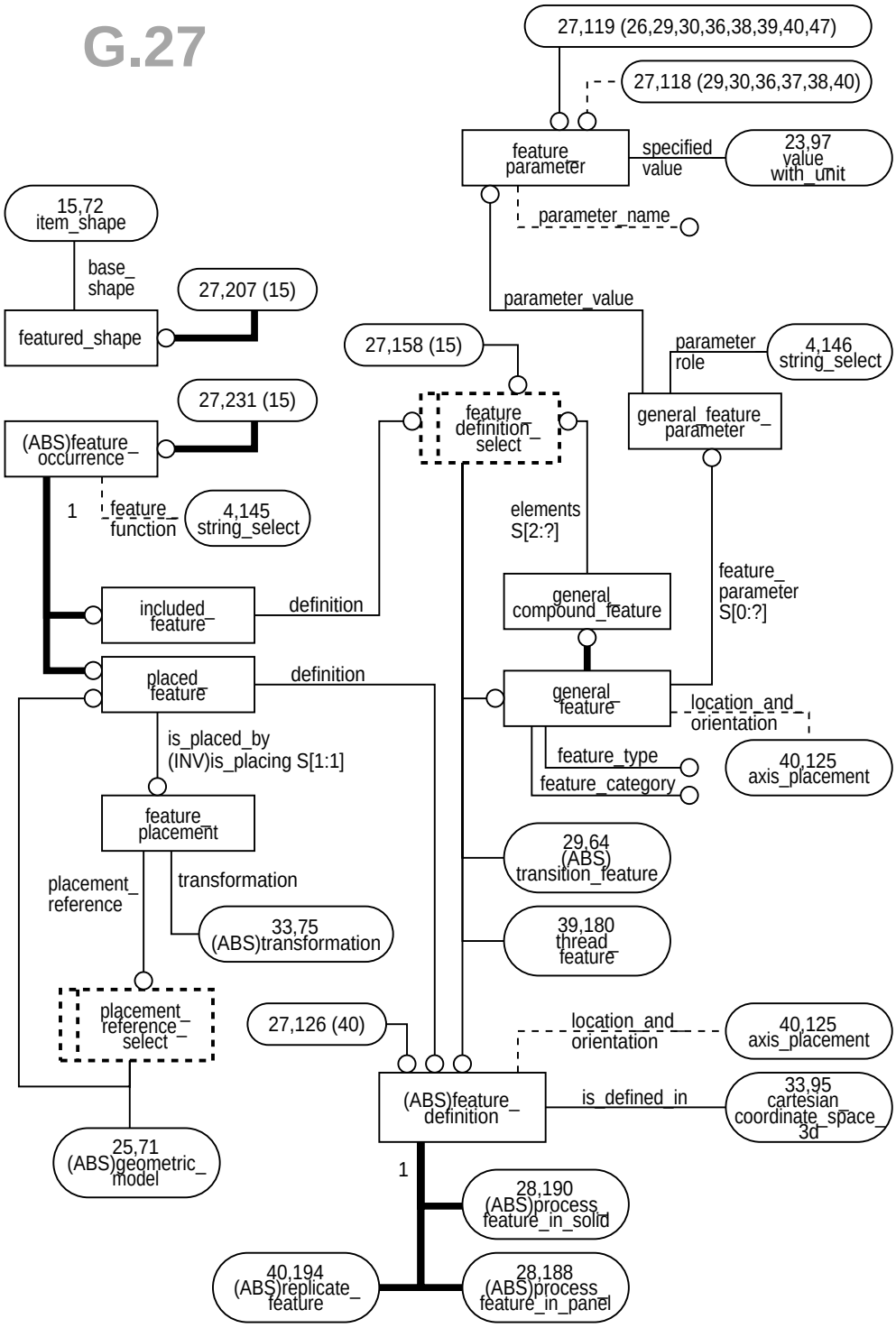


Figure G.27 – ARM diagram in EXPRESS_G: 27 of 50

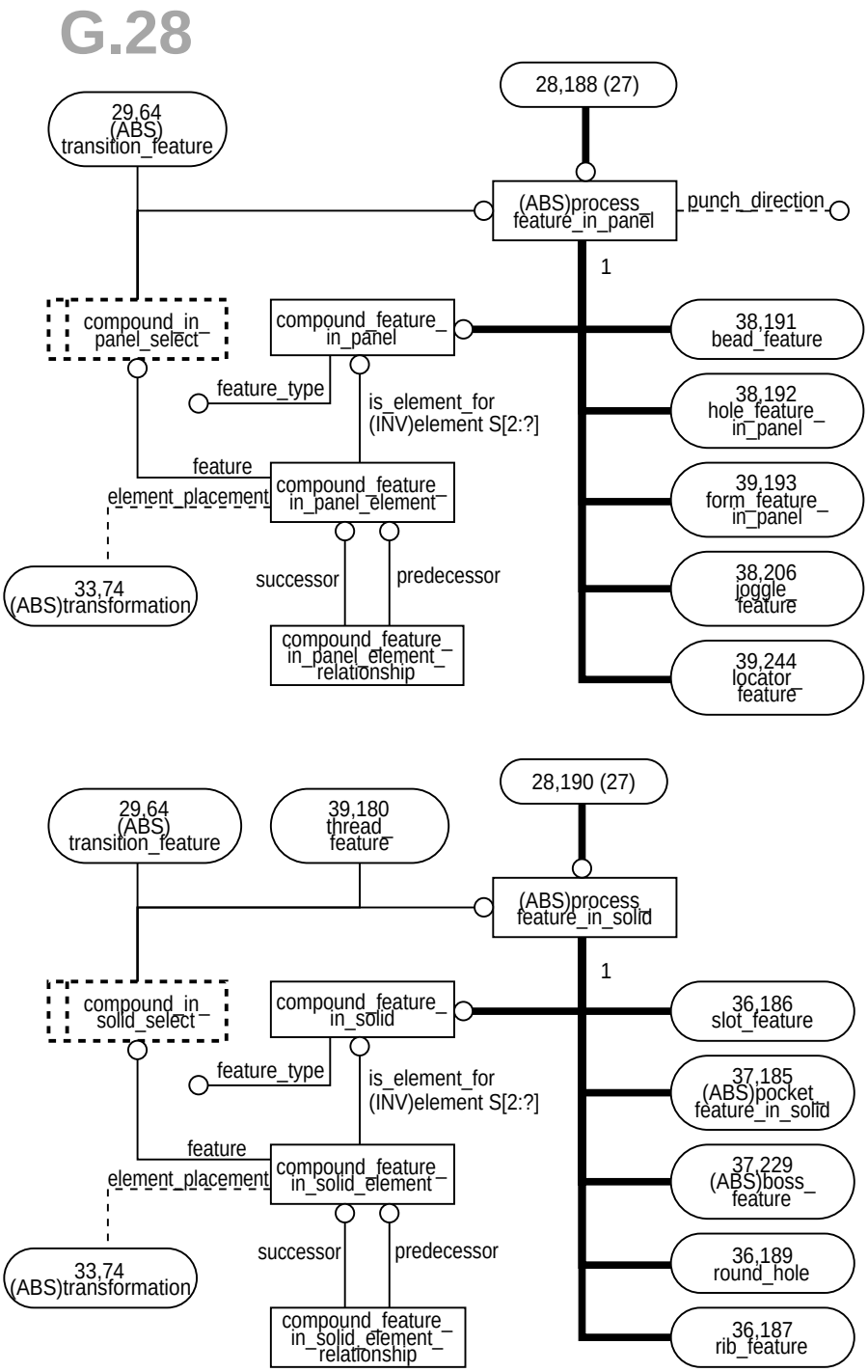


Figure G.28 – ARM diagram in EXPRESS_G: 28 of 50

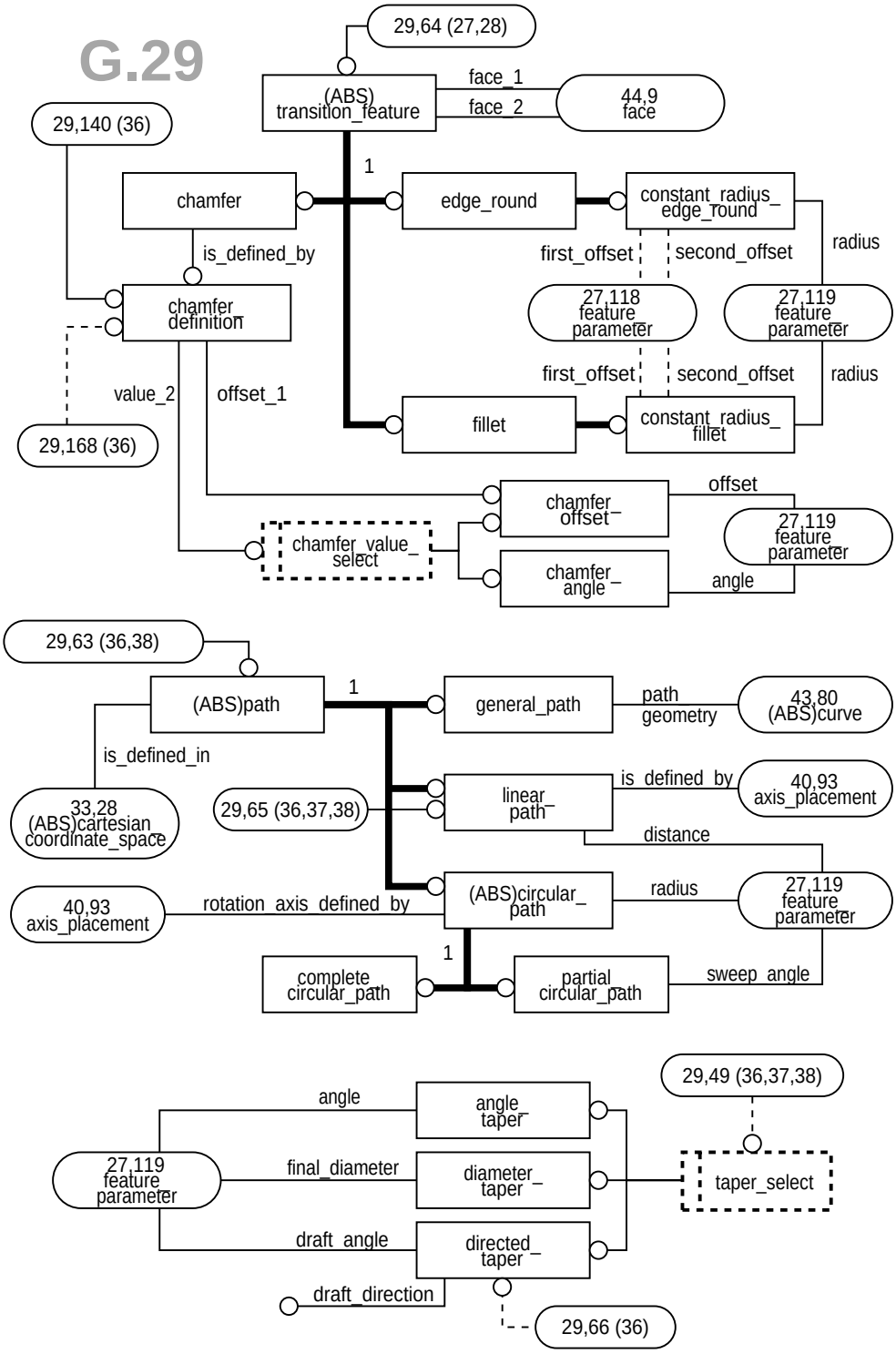


Figure G.29 – ARM diagram in EXPRESS_G: 29 of 50

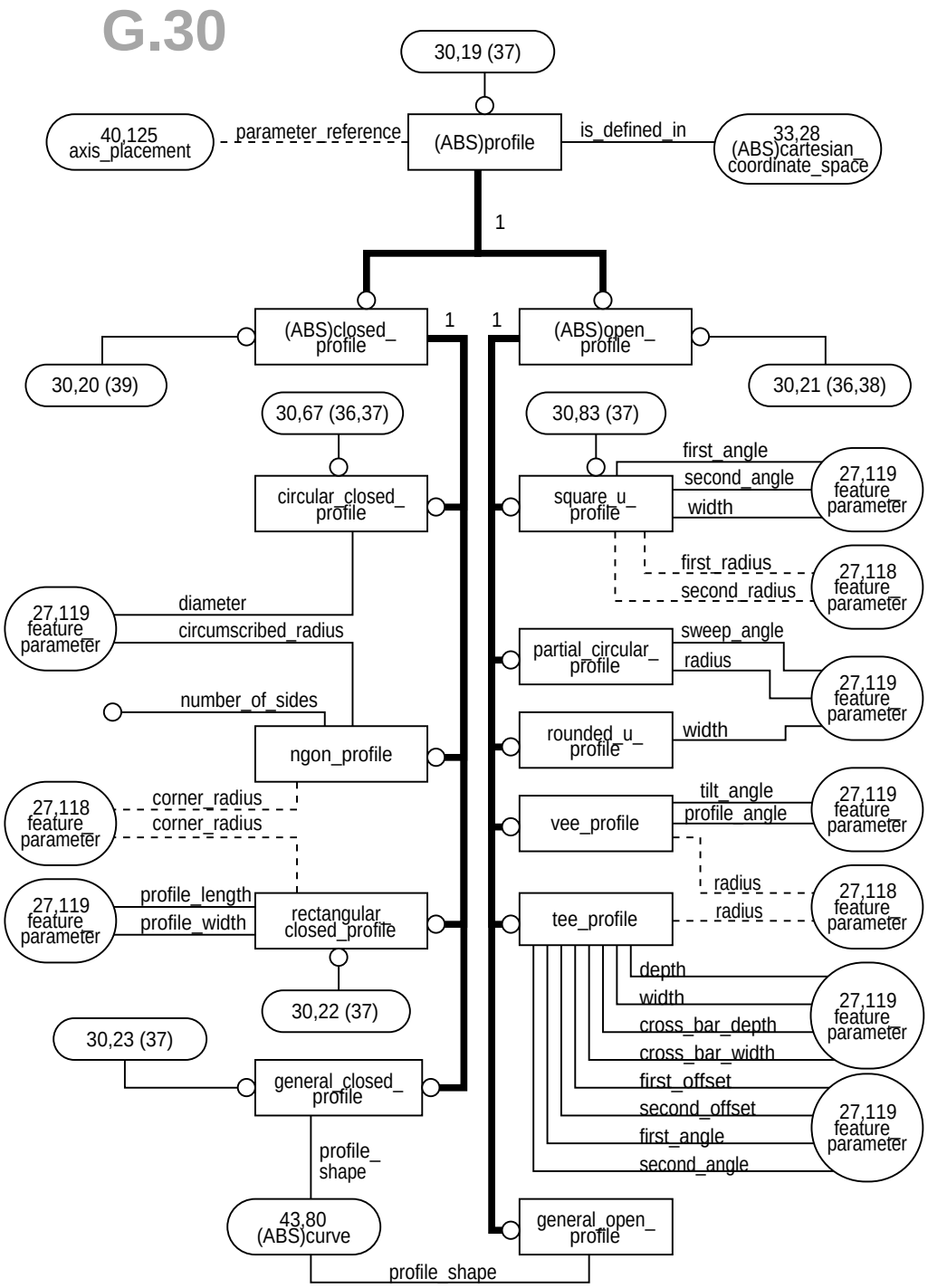


Figure G.30 – ARM diagram in EXPRESS_G: 30 of 50

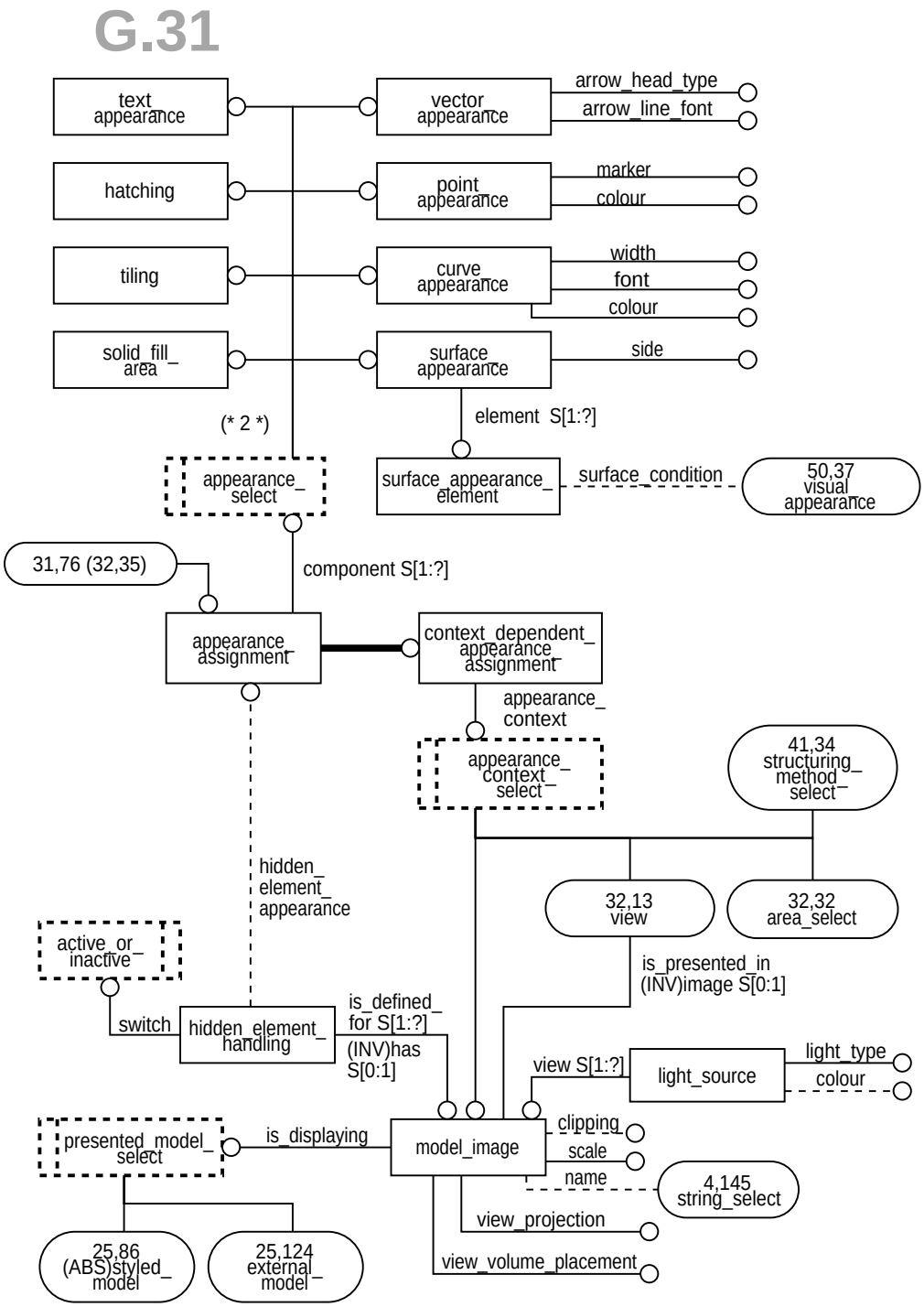


Figure G.31 – ARM diagram in EXPRESS_G: 31 of 50

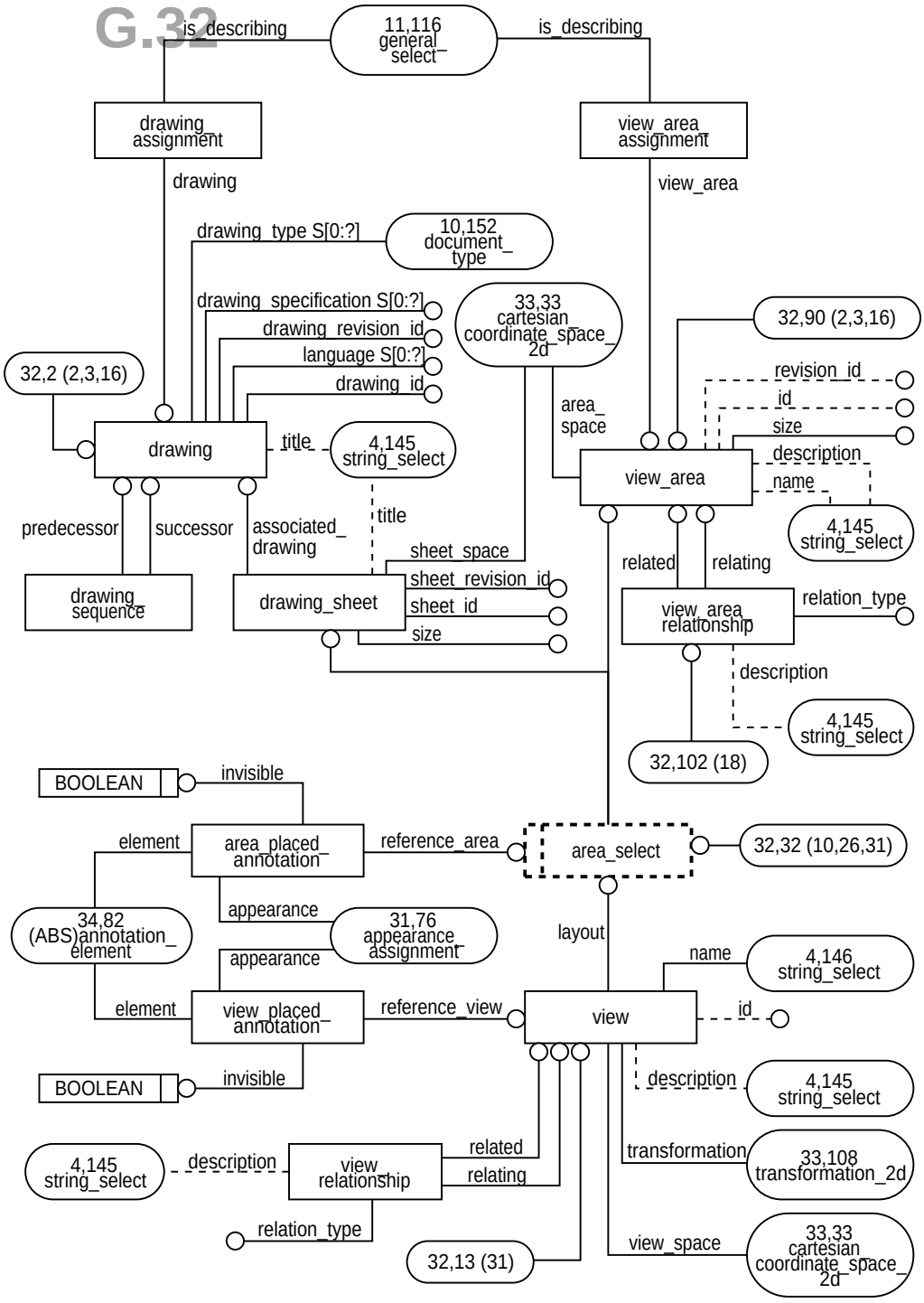


Figure G.32 – ARM diagram in EXPRESS_G: 32 of 50

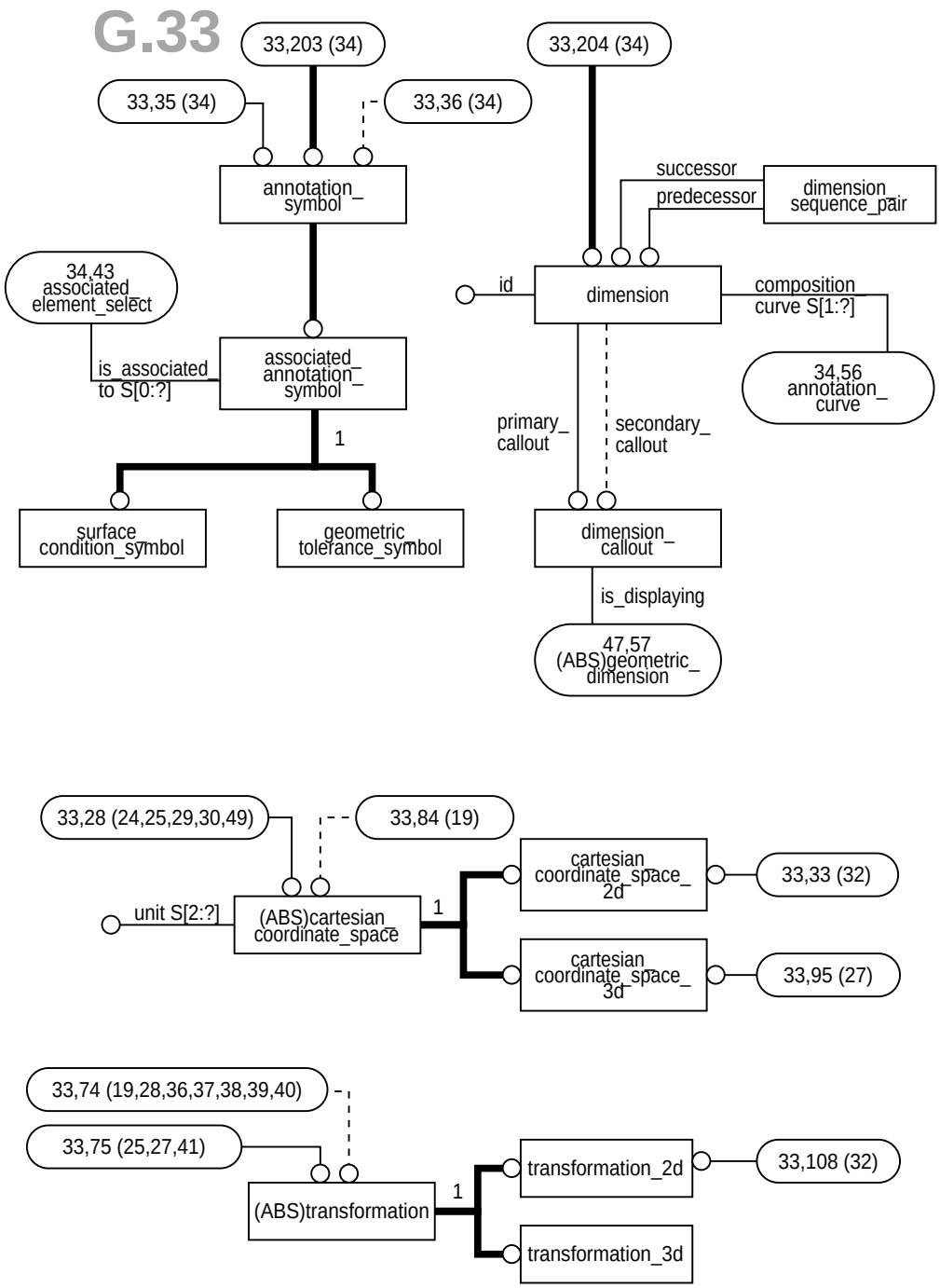


Figure G.33 – ARM diagram in EXPRESS_G: 33 of 50

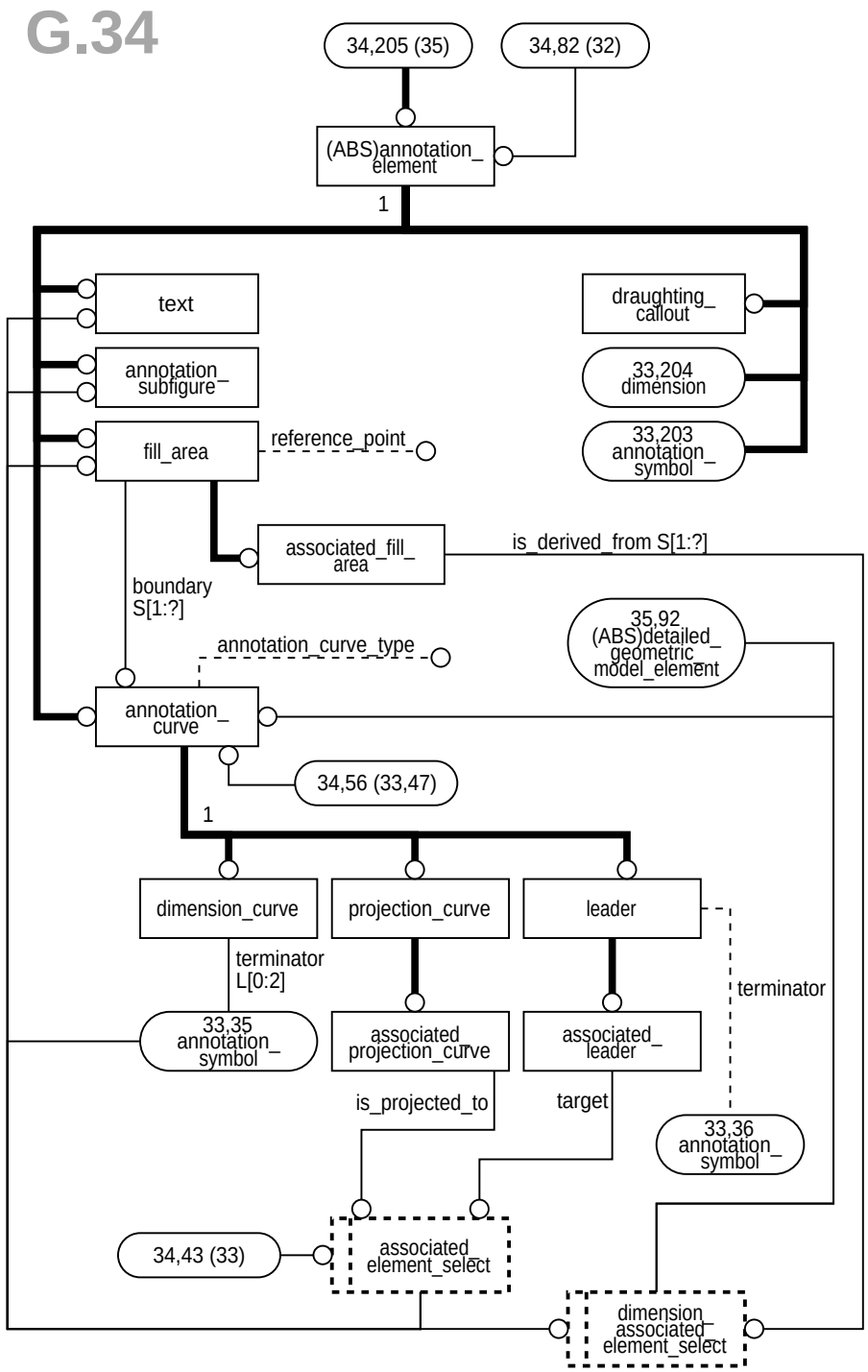


Figure G.34 – ARM diagram in EXPRESS_G: 34 of 50

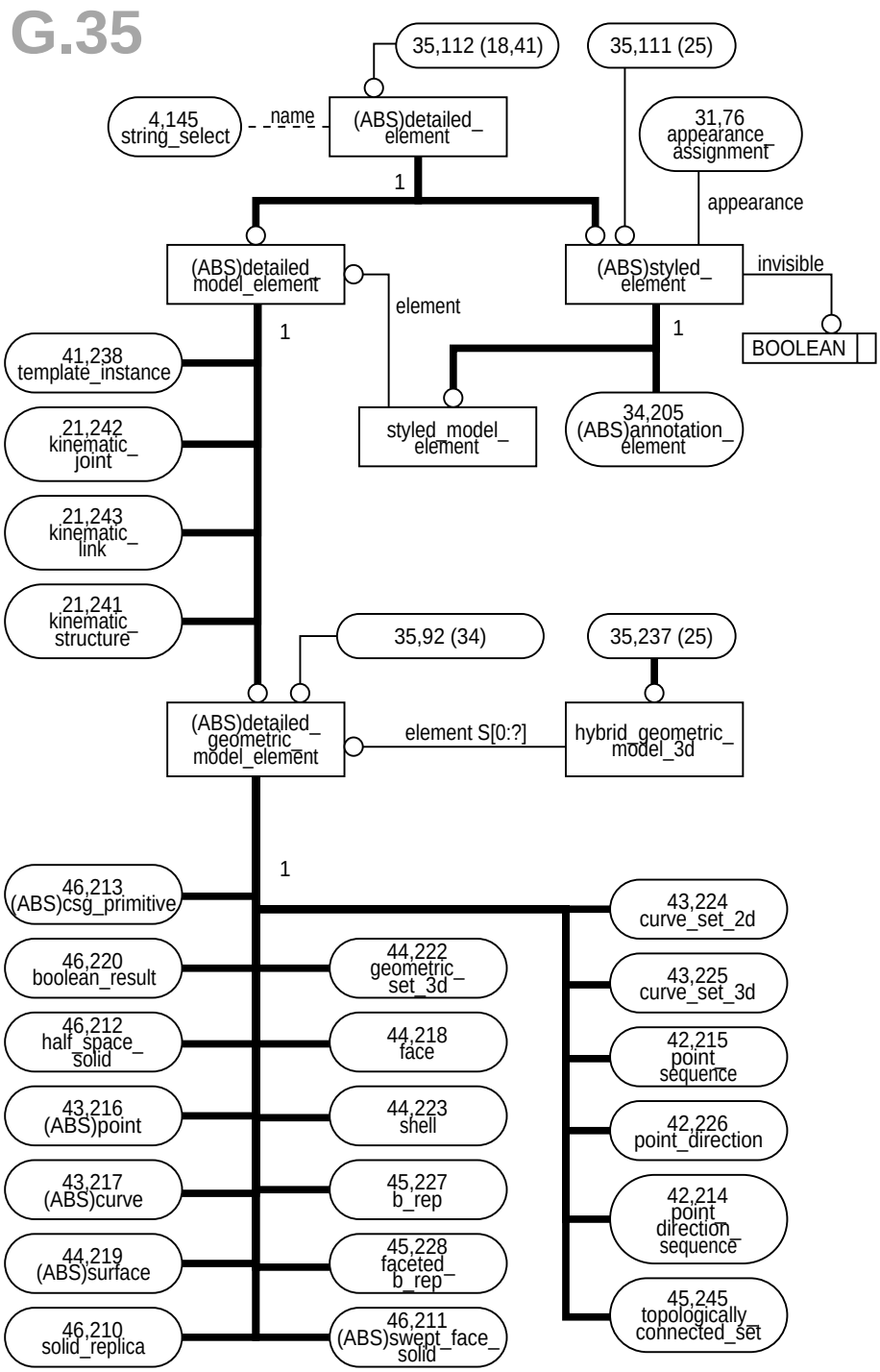


Figure G.35 – ARM diagram in EXPRESS_G: 35 of 50

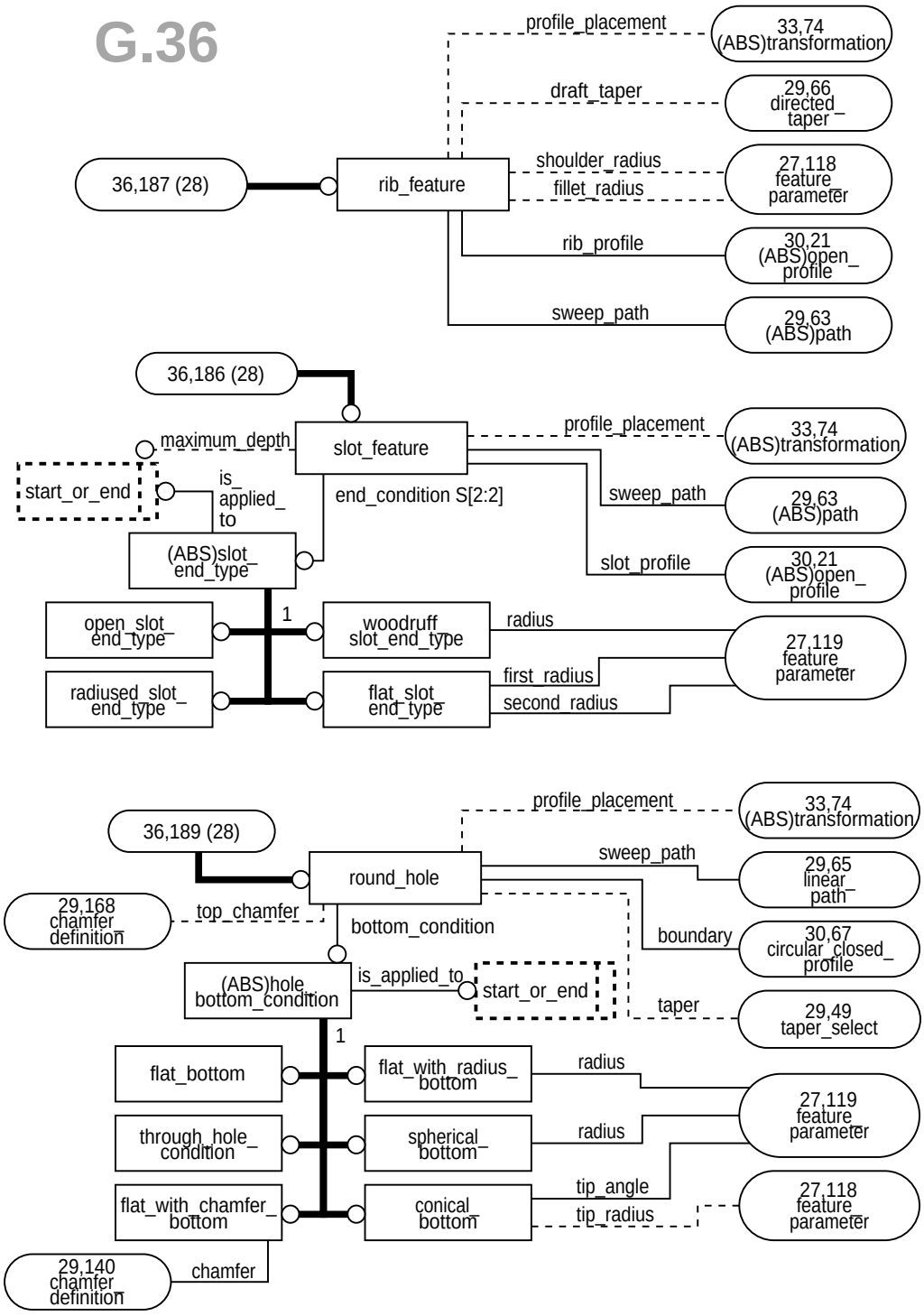


Figure G.36 – ARM diagram in EXPRESS_G: 36 of 50

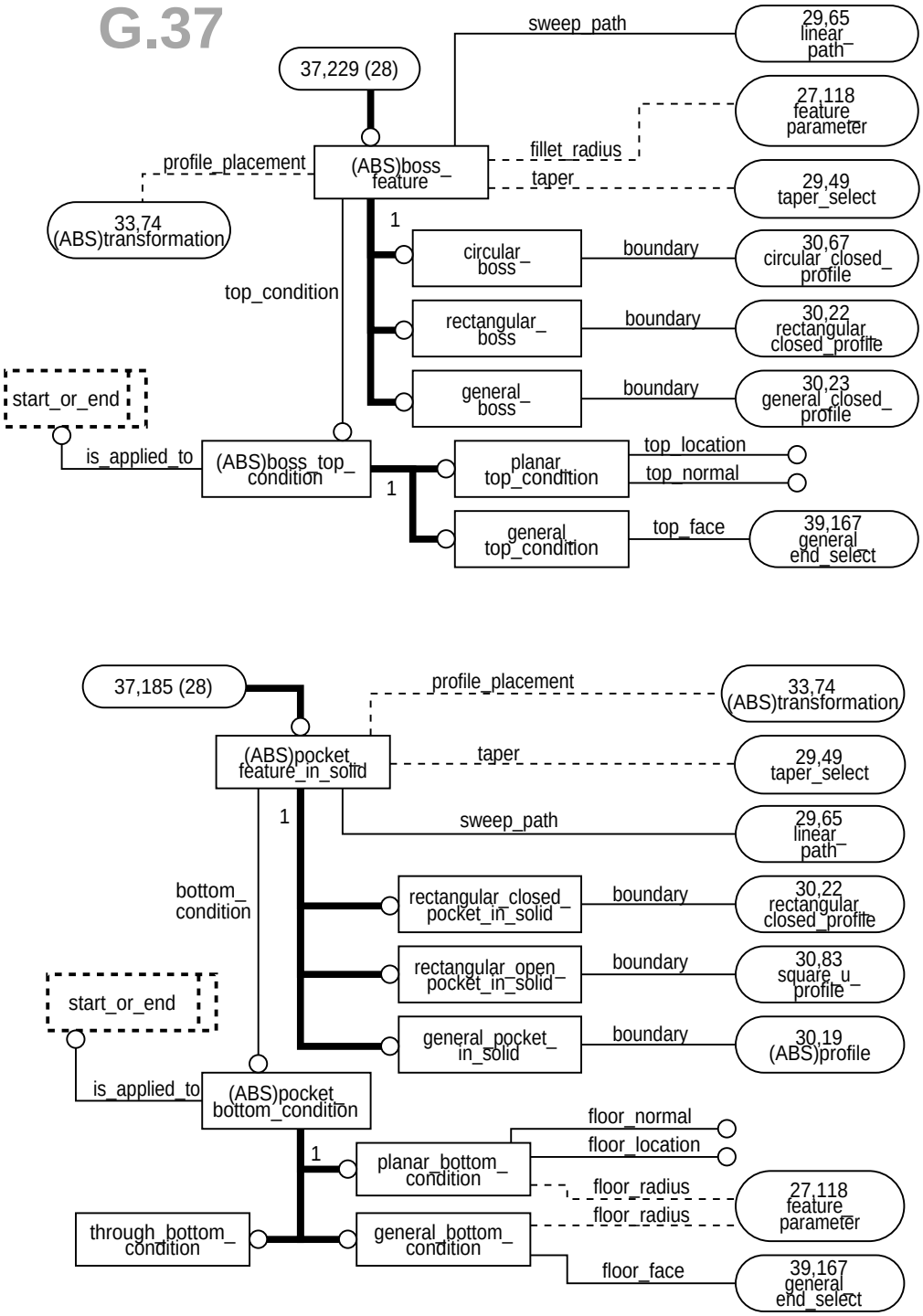


Figure G.37 – ARM diagram in EXPRESS_G: 37 of 50

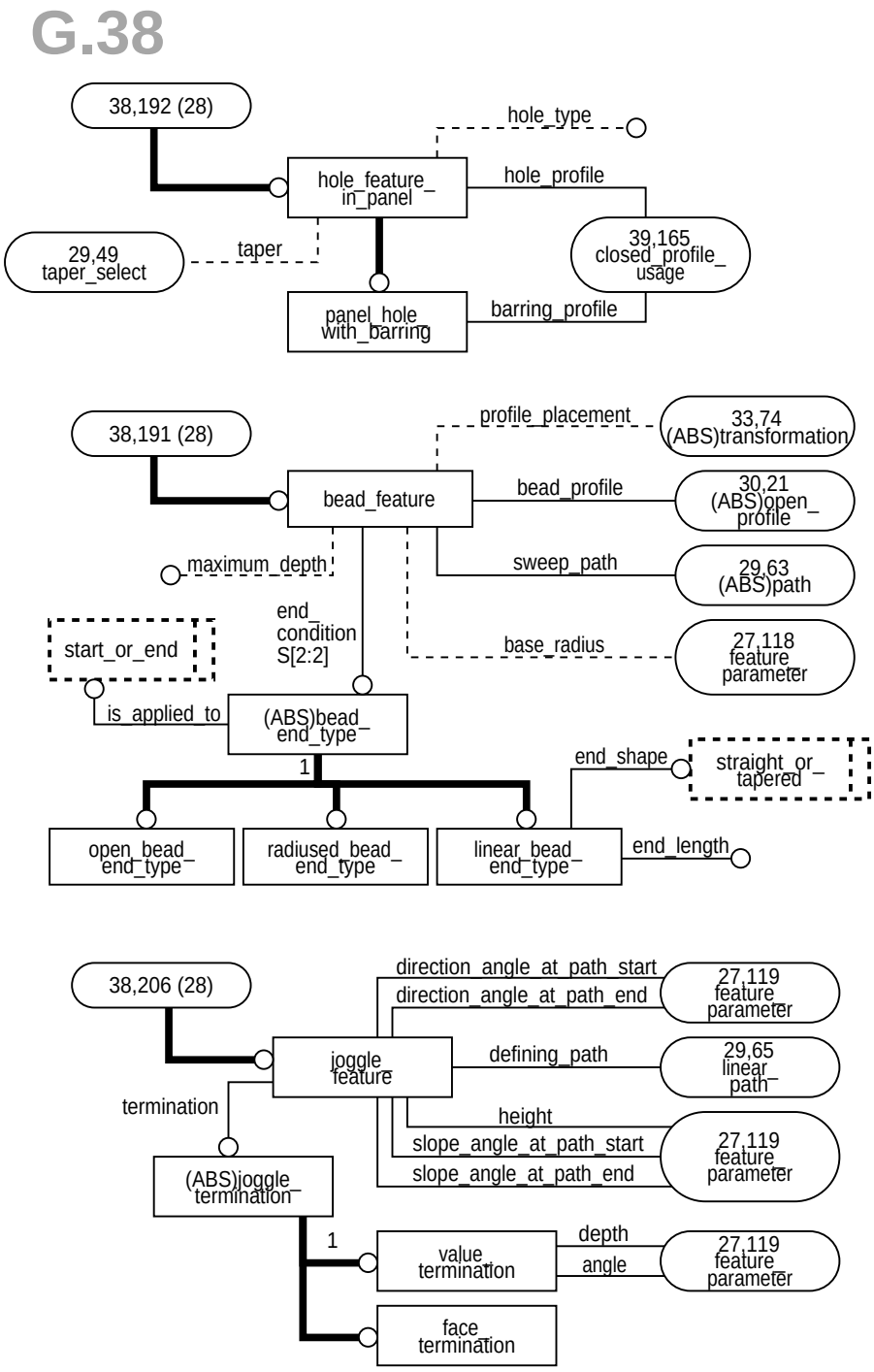


Figure G.38 – ARM diagram in EXPRESS_G: 38 of 50

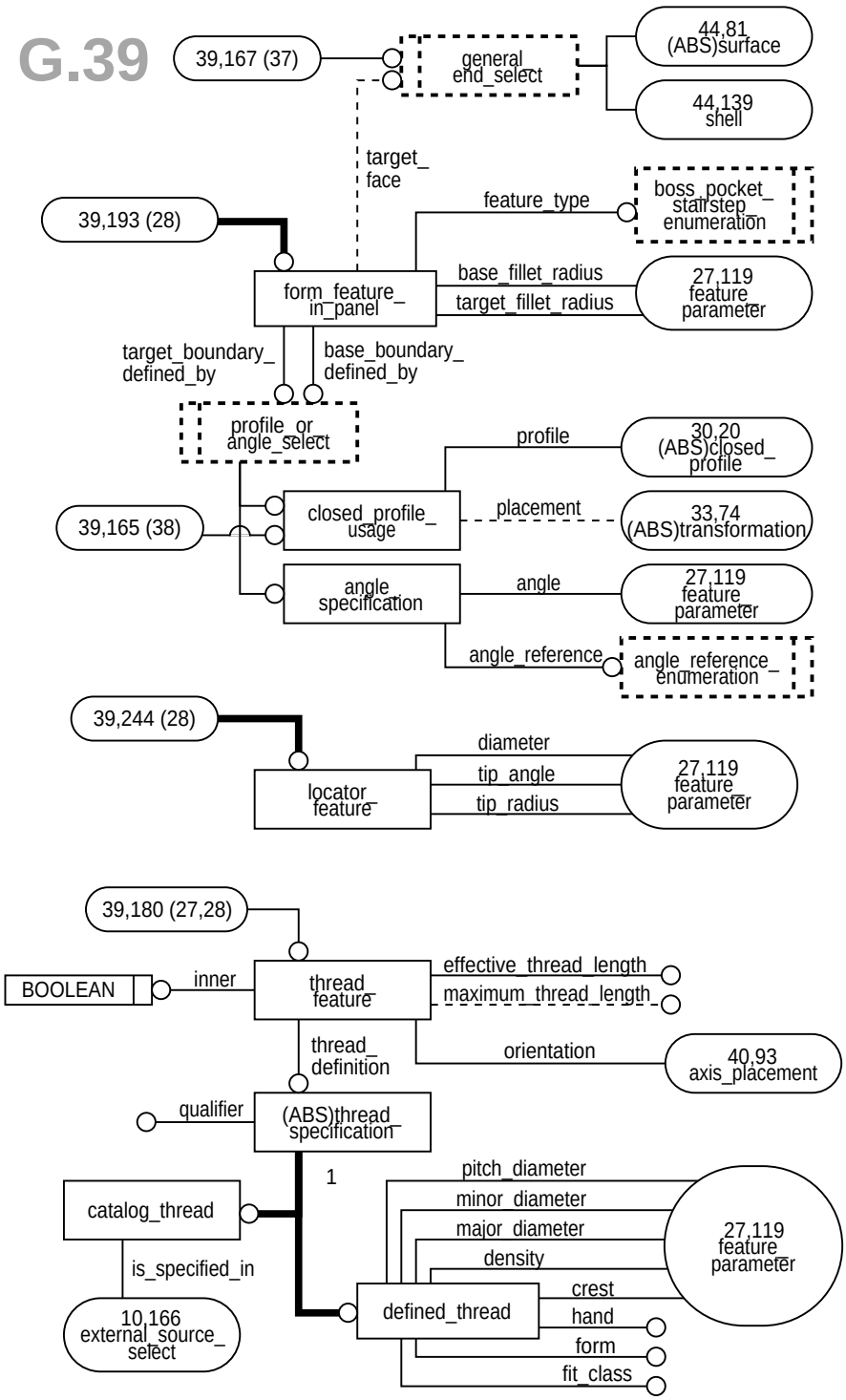


Figure G.39 – ARM diagram in EXPRESS_G: 39 of 50

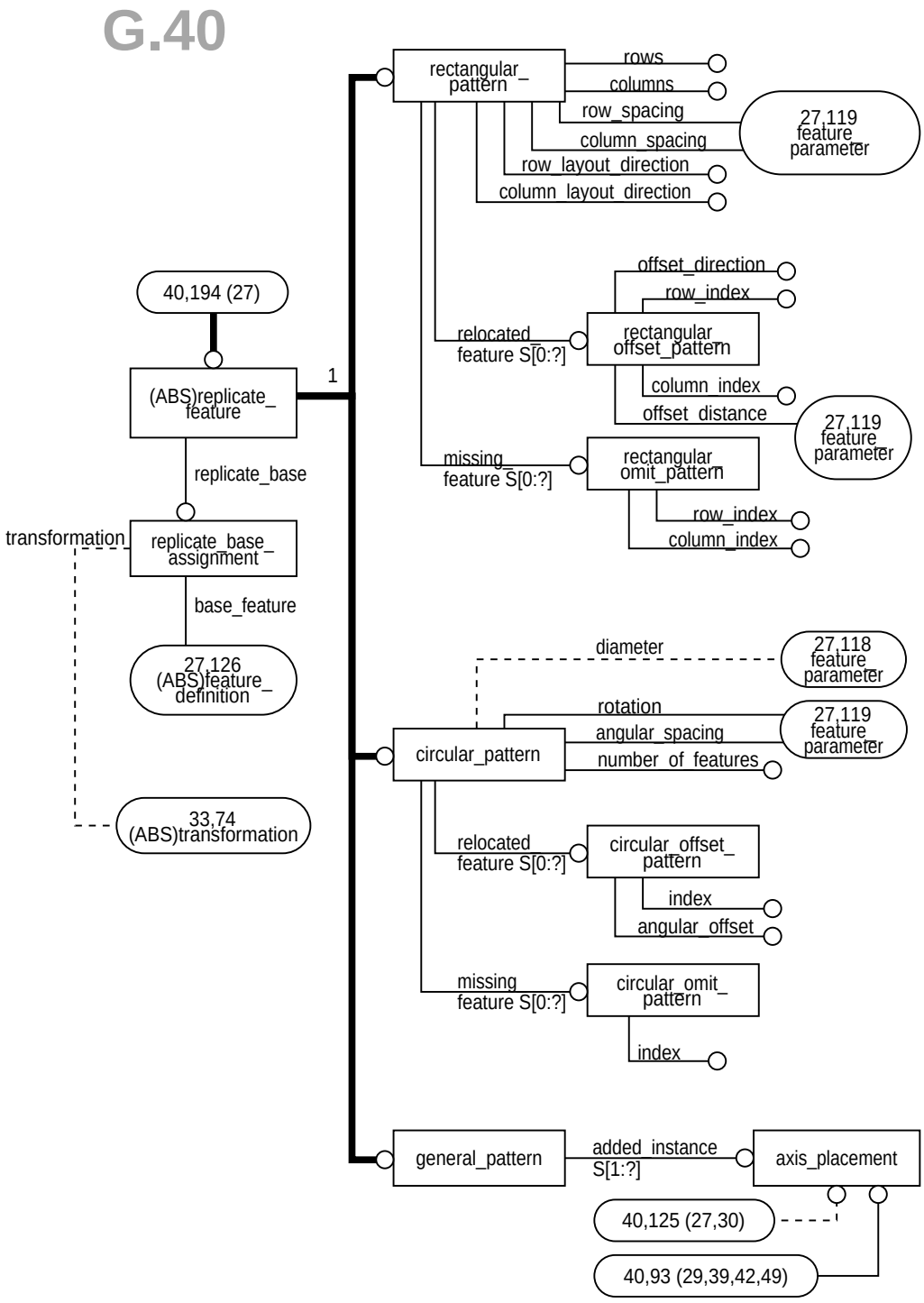


Figure G.40 – ARM diagram in EXPRESS_G: 40 of 50

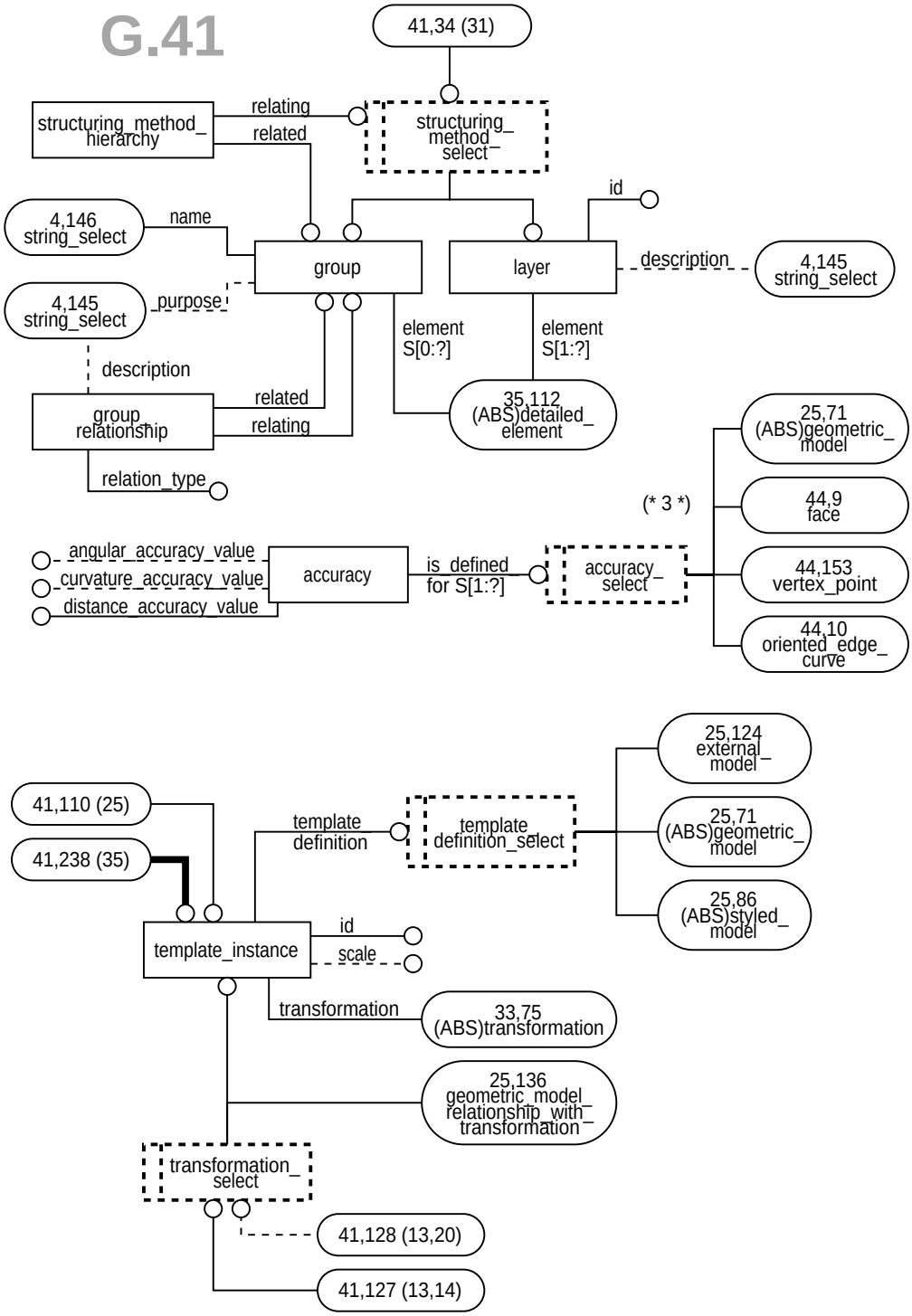


Figure G.41 – ARM diagram in EXPRESS_G: 41 of 50

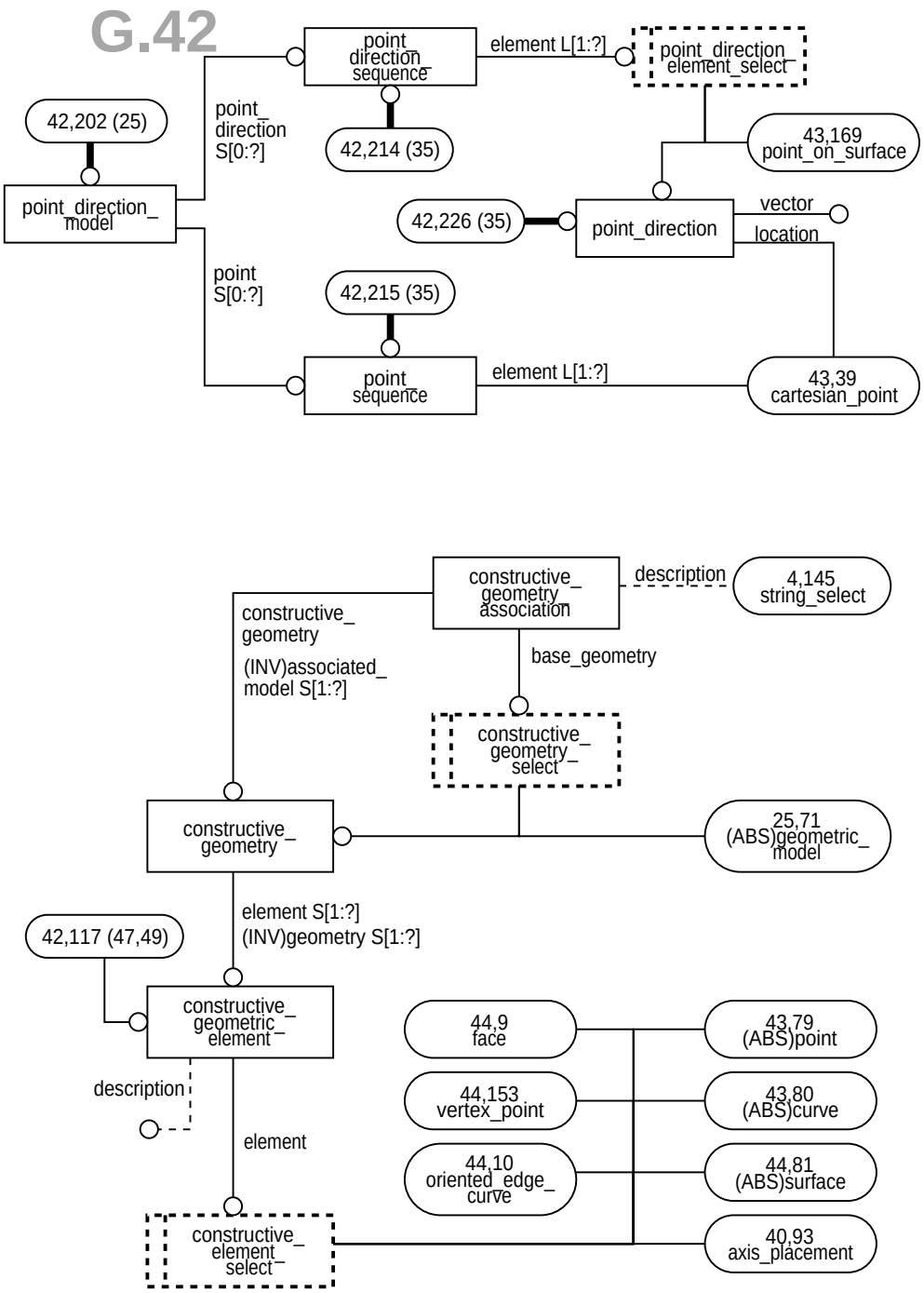


Figure G.42 – ARM diagram in EXPRESS_G: 42 of 50

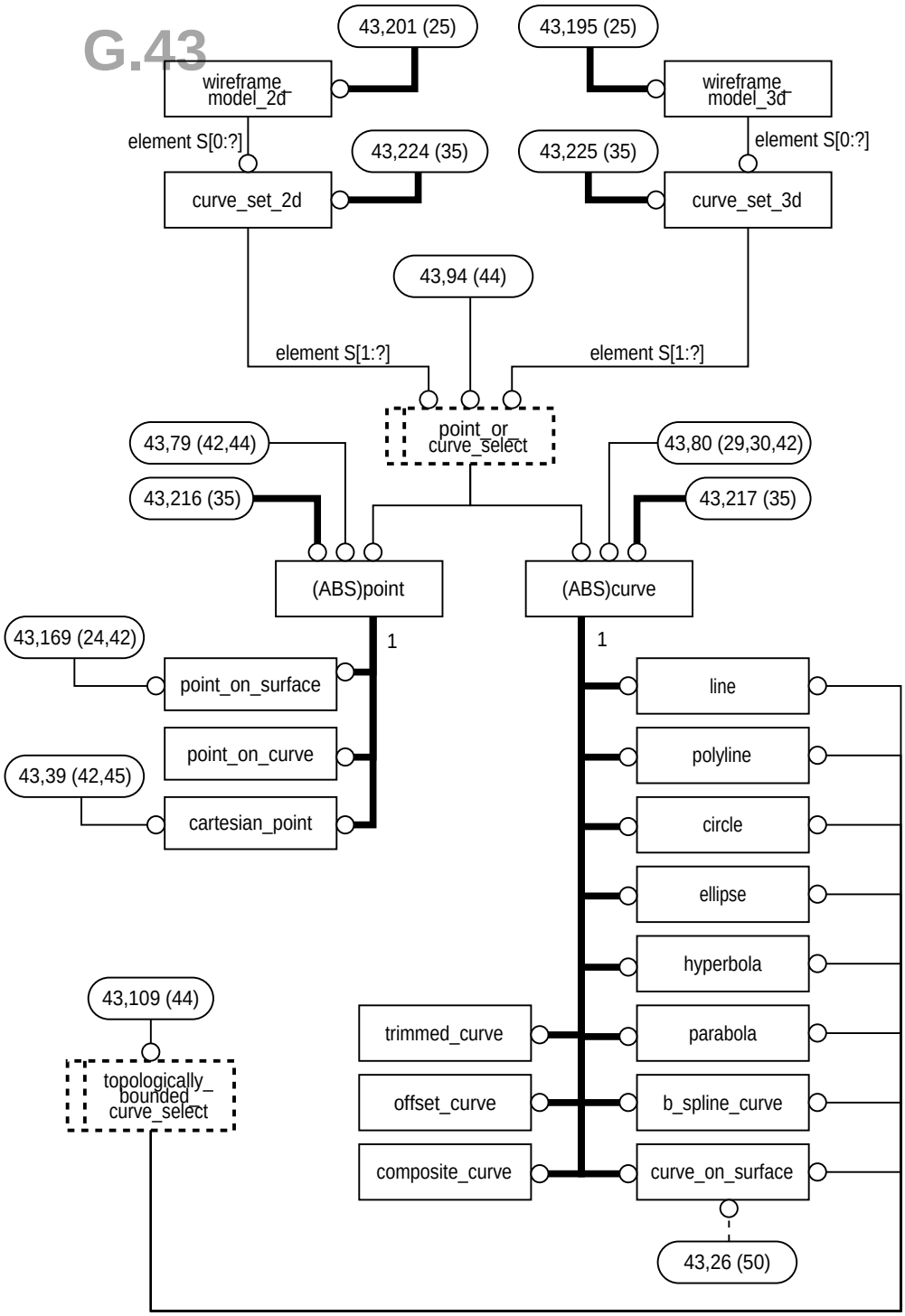


Figure G.43 – ARM diagram in EXPRESS_G: 43 of 50

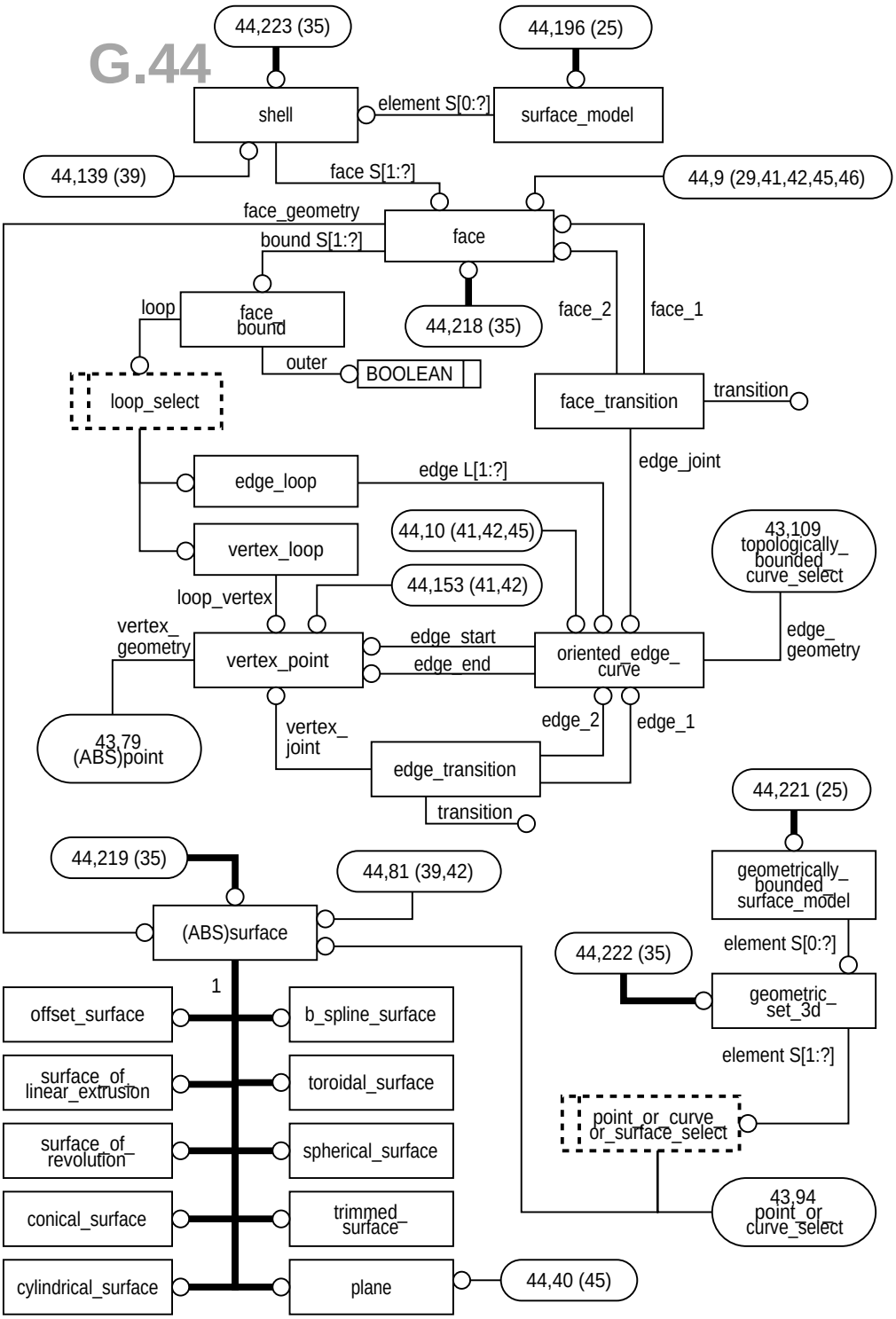


Figure G.44 – ARM diagram in EXPRESS_G: 44 of 50

G.45

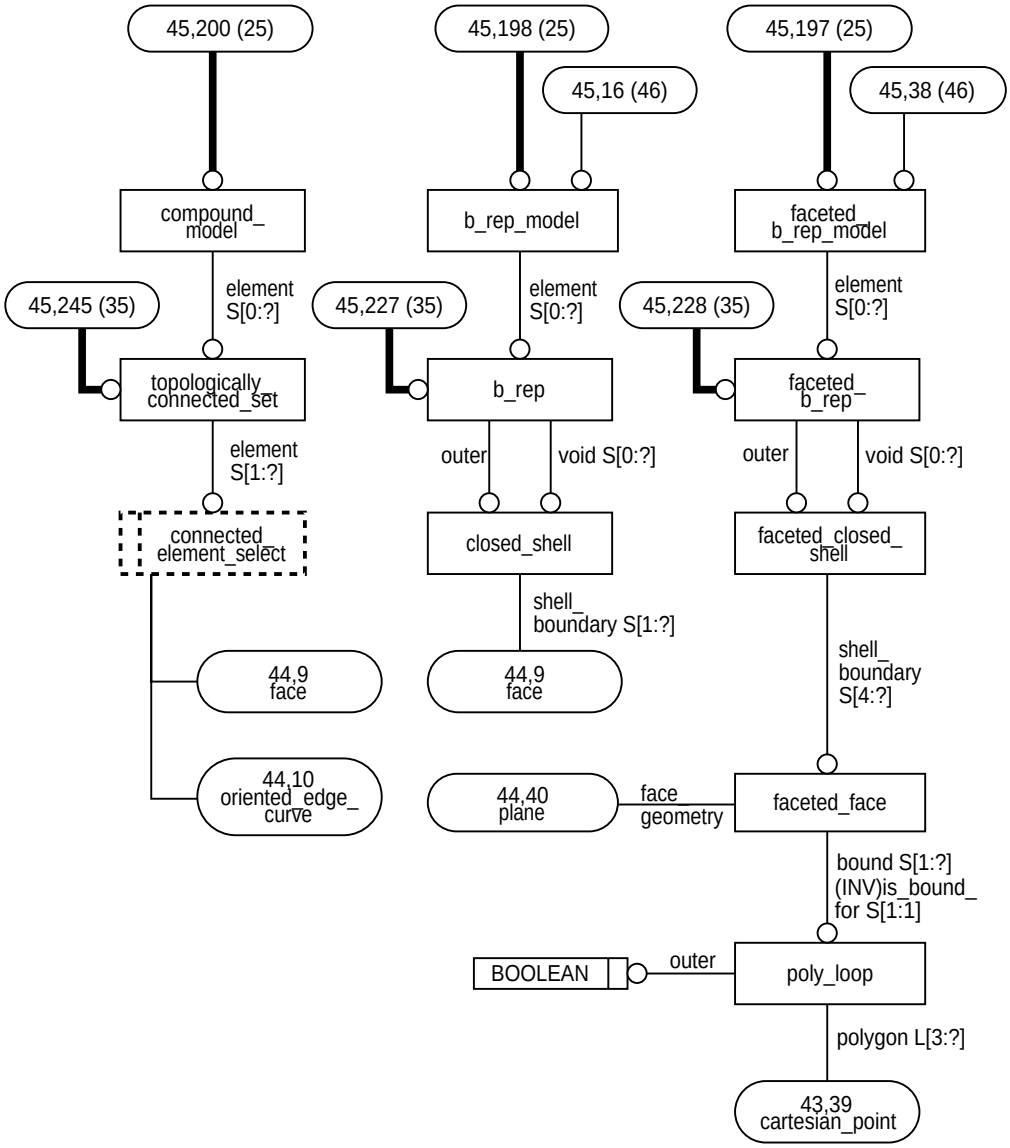


Figure G.45 – ARM diagram in EXPRESS_G: 45 of 50

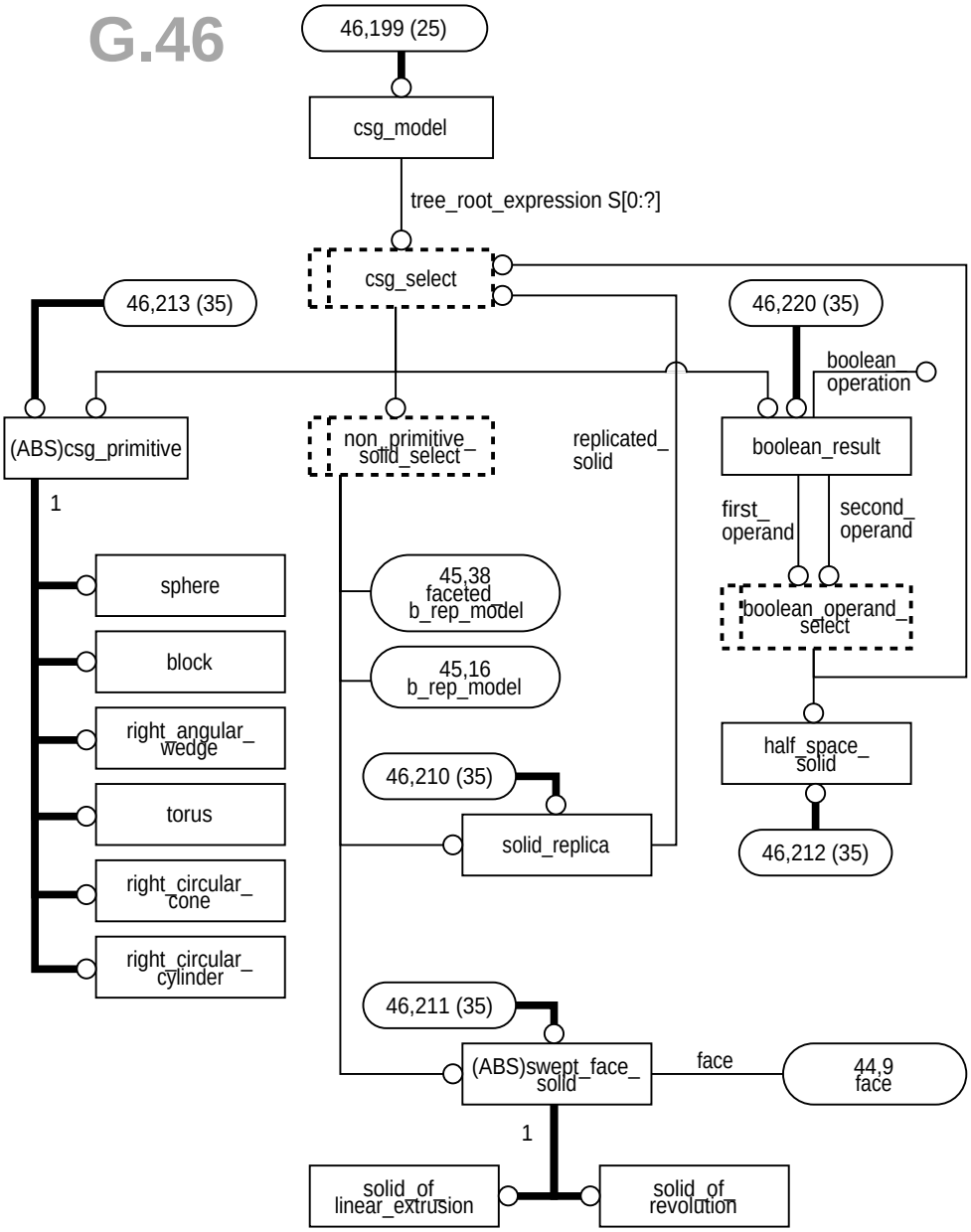


Figure G.46 – ARM diagram in EXPRESS_G: 46 of 50

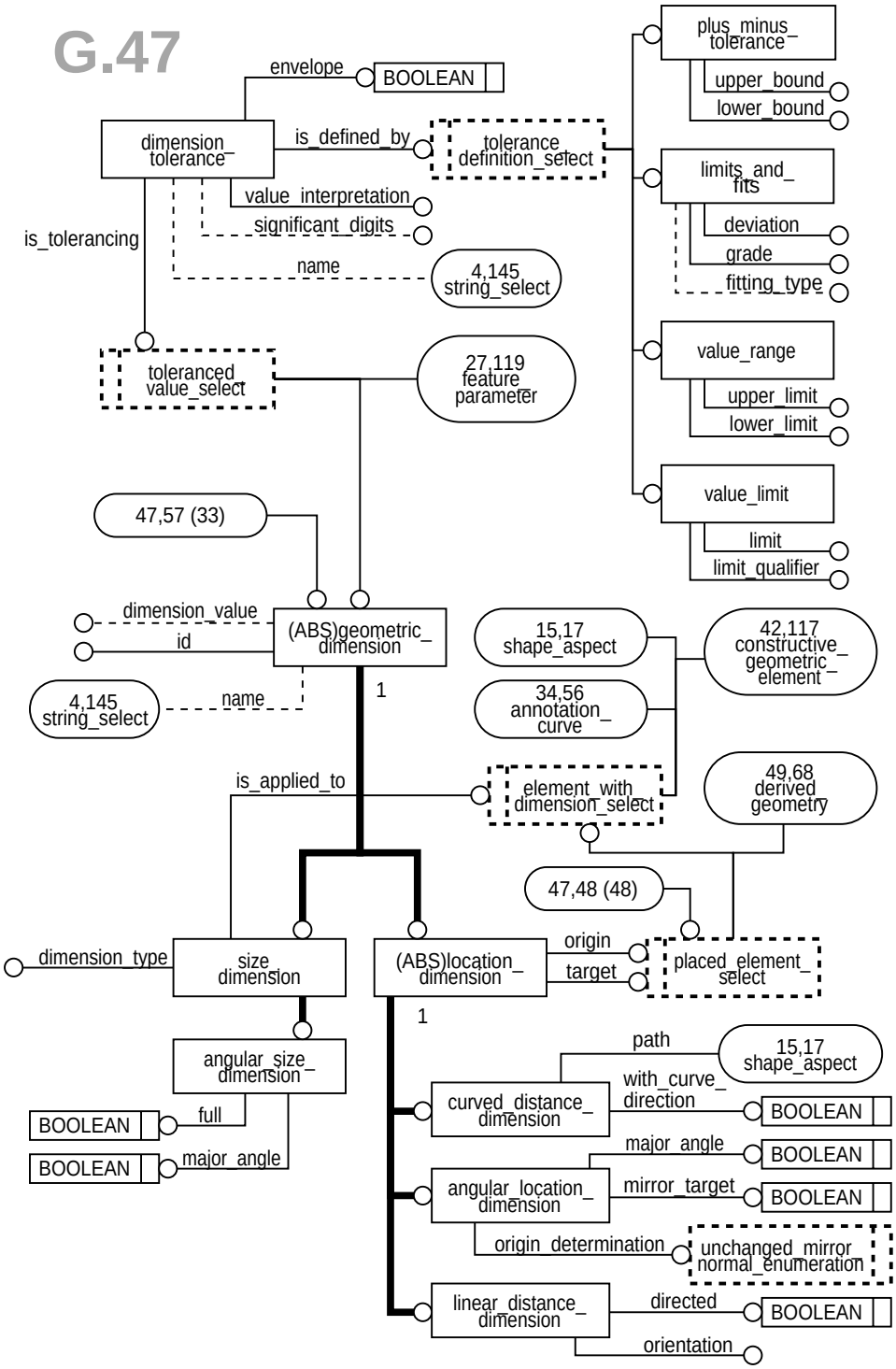


Figure G.47 – ARM diagram in EXPRESS_G: 47 of 50

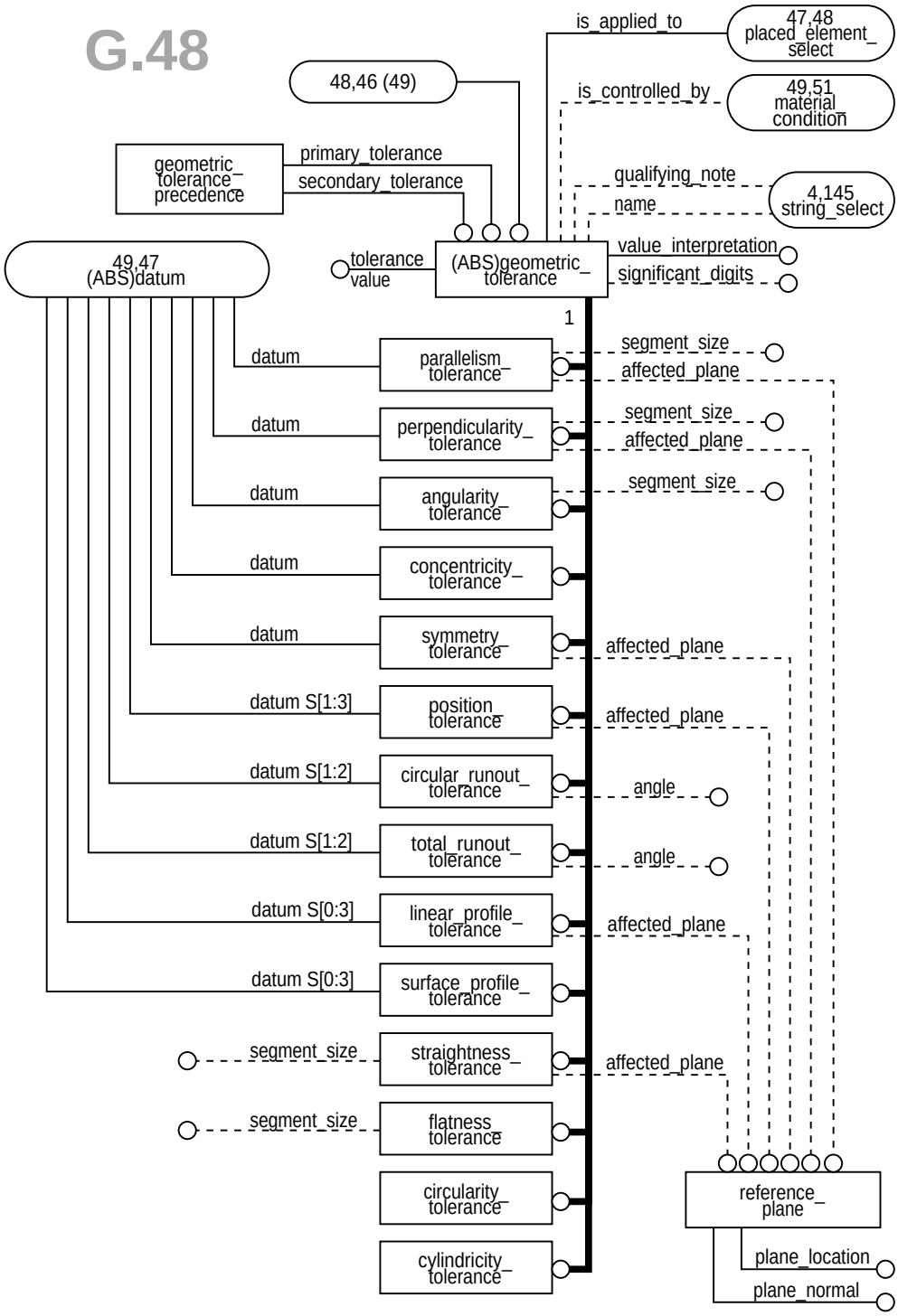


Figure G.48 – ARM diagram in EXPRESS_G: 48 of 50

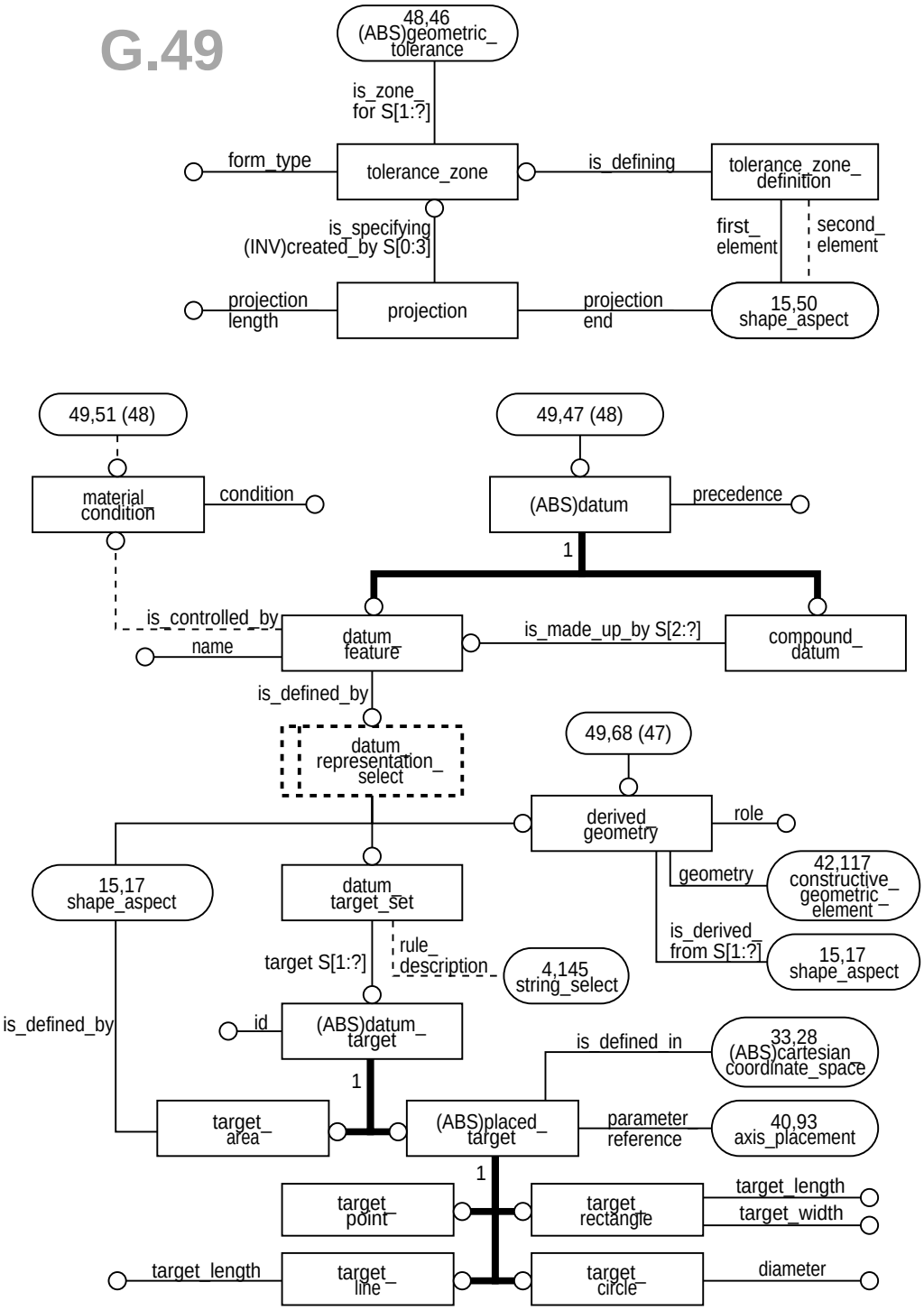


Figure G.49 – ARM diagram in EXPRESS_G: 49 of 50

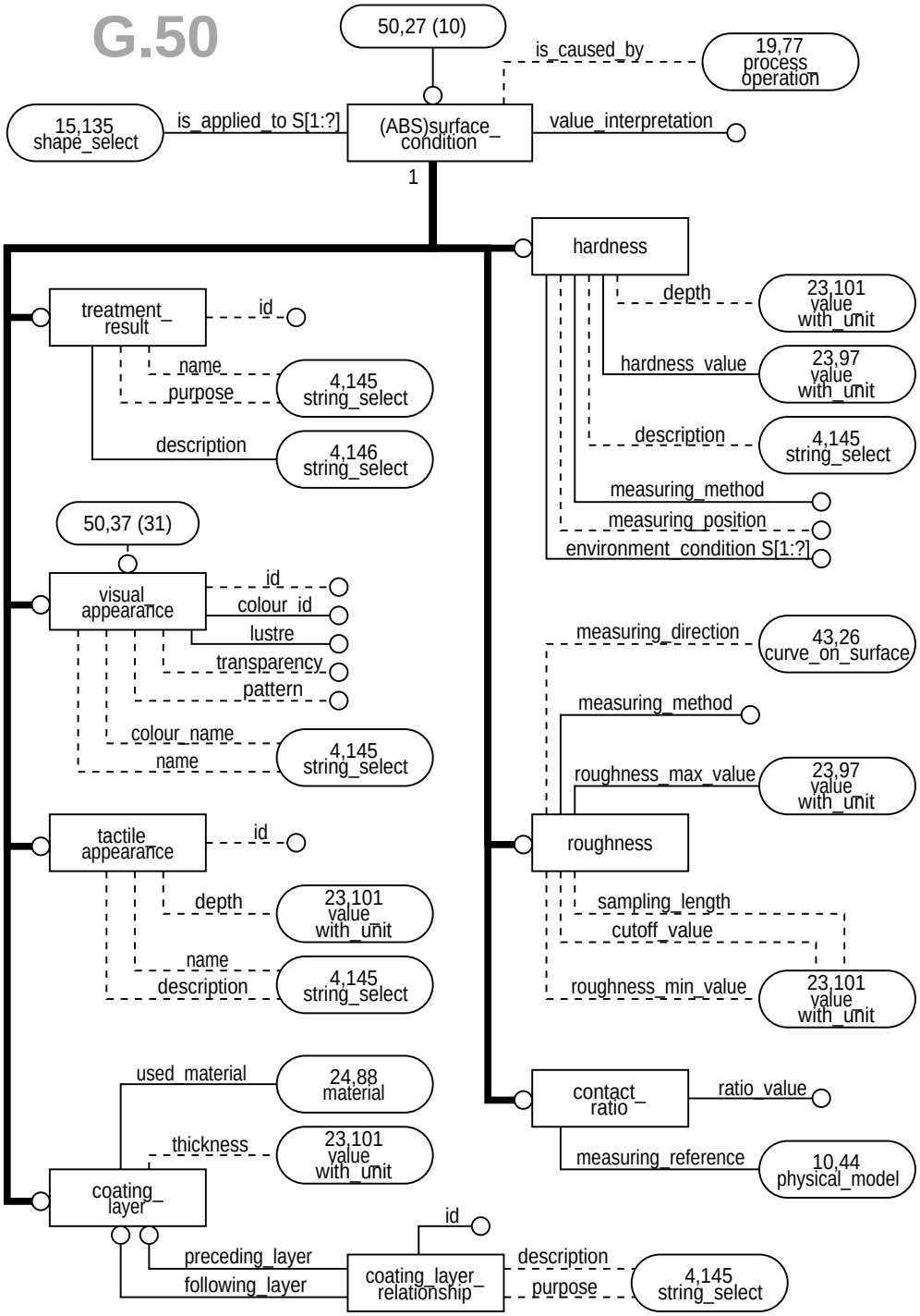


Figure G.50 – ARM diagram in EXPRESS_G: 50 of 50

NOTE 3 – The following footnotes correspond to the footnote marks in the ARM EXPRESS-G figures above:

1 Each Default_setting_association has default_value defined by zero or one object of each kind included in the default_setting_select.

EXAMPLE 242 – There shall not be two objects of Default_chamfer associated to one Default_-setting_association.

2 Each object included in the appearance_select is related to one or more Appearance_assignment object.

3 Each object included in the accuracy_select is related to zero or one Accuracy object.

Annex H
(informative)

AIM EXPRESS-G

The following diagrams correspond to the AIM EXPRESS annotated listing given in annex A. The diagrams use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in annex D of ISO 10303-11.

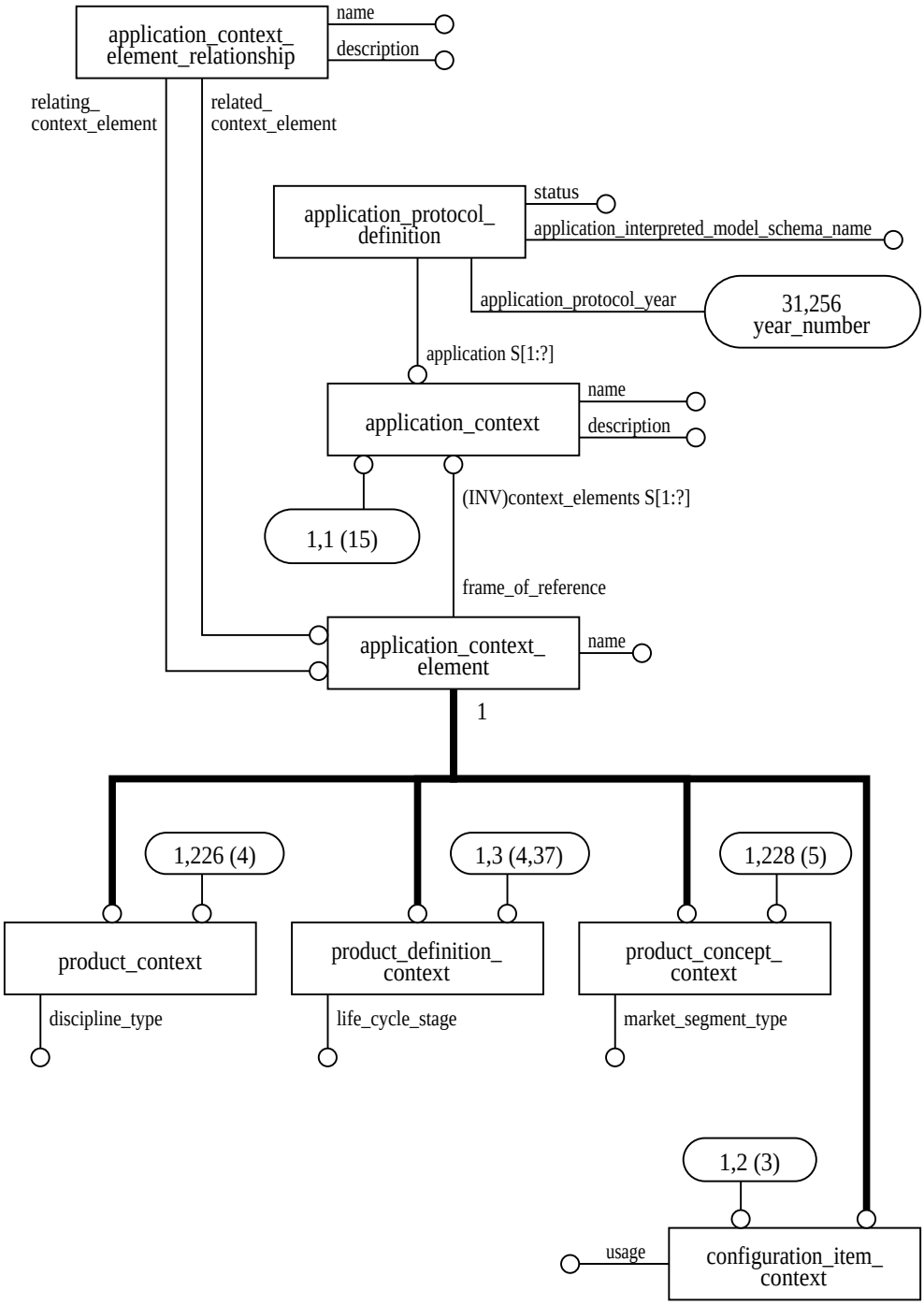


Figure H.1 – AIM diagram in EXPRESS_G: 1 of 124

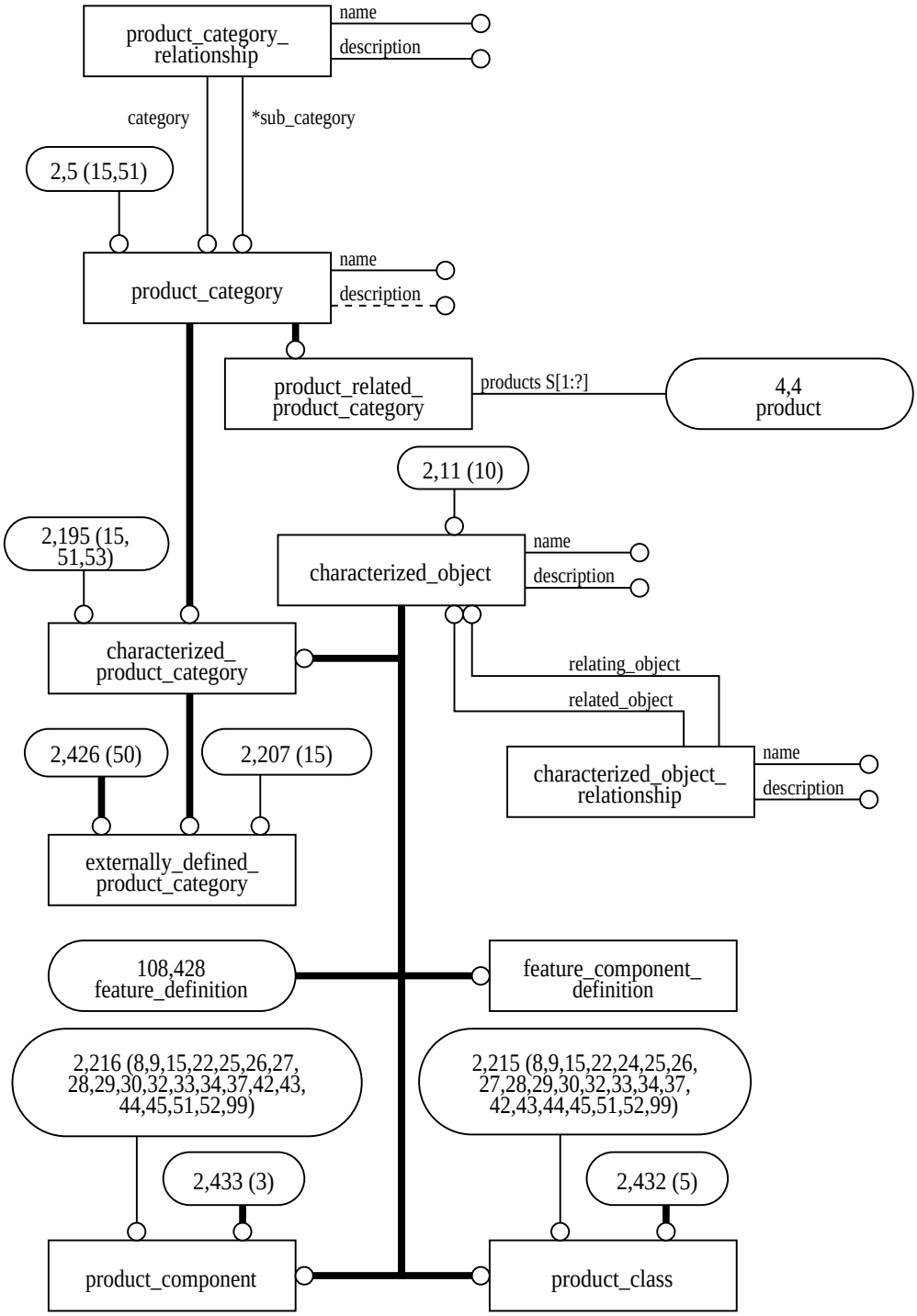


Figure H.2 – AIM diagram in EXPRESS_G: 2 of 124

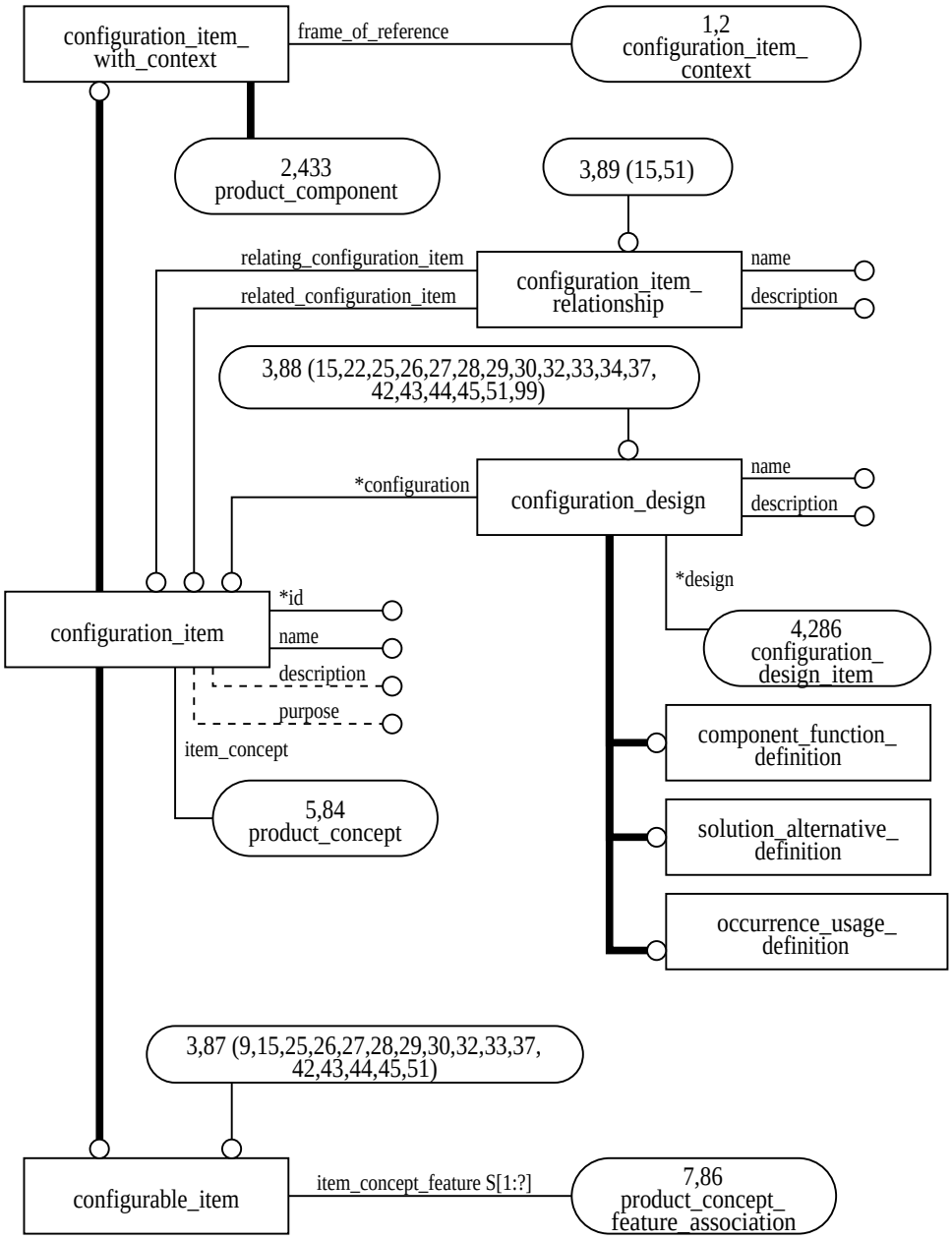


Figure H.3 – AIM diagram in EXPRESS_G: 3 of 124

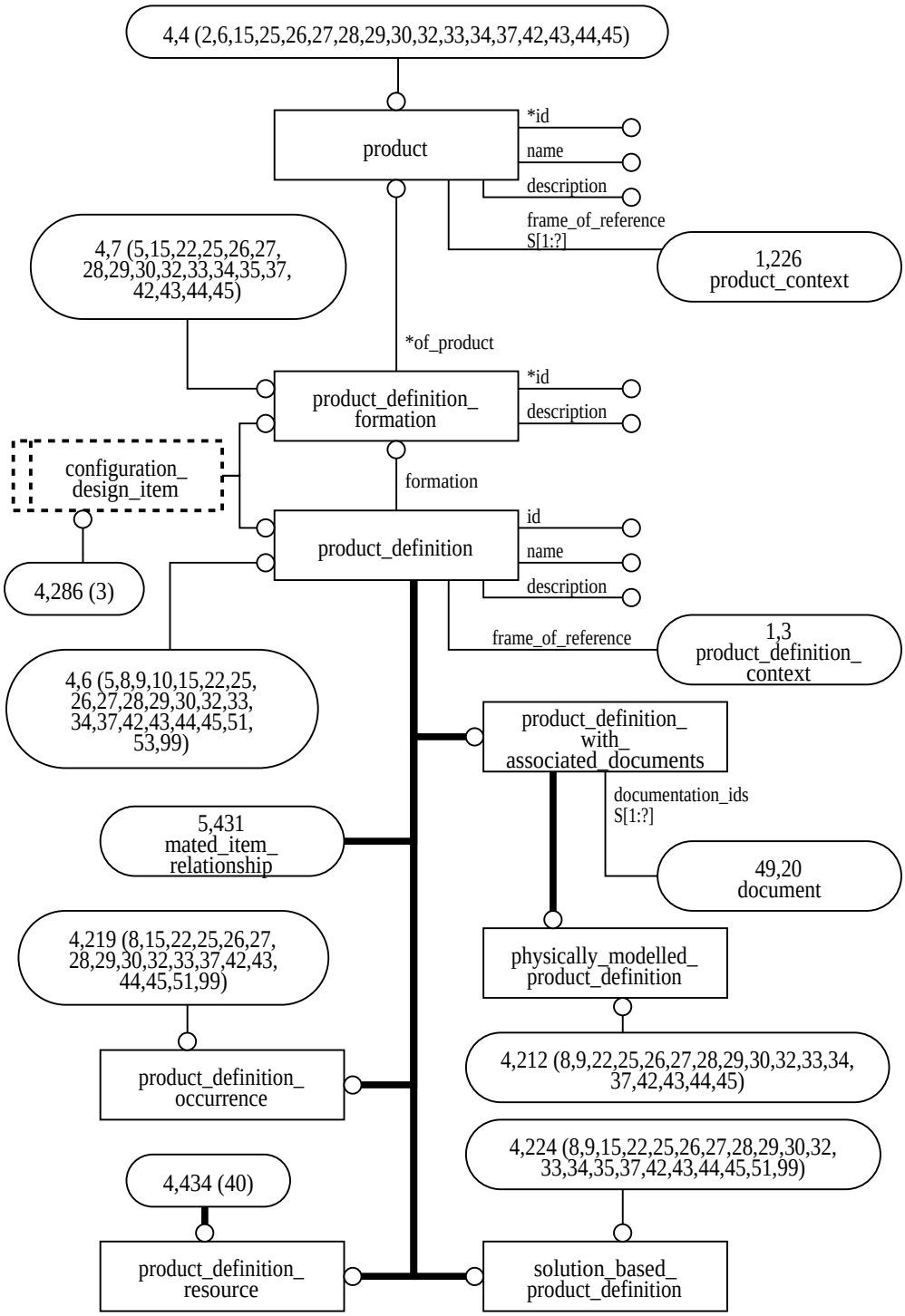


Figure H.4 – AIM diagram in EXPRESS_G: 4 of 124

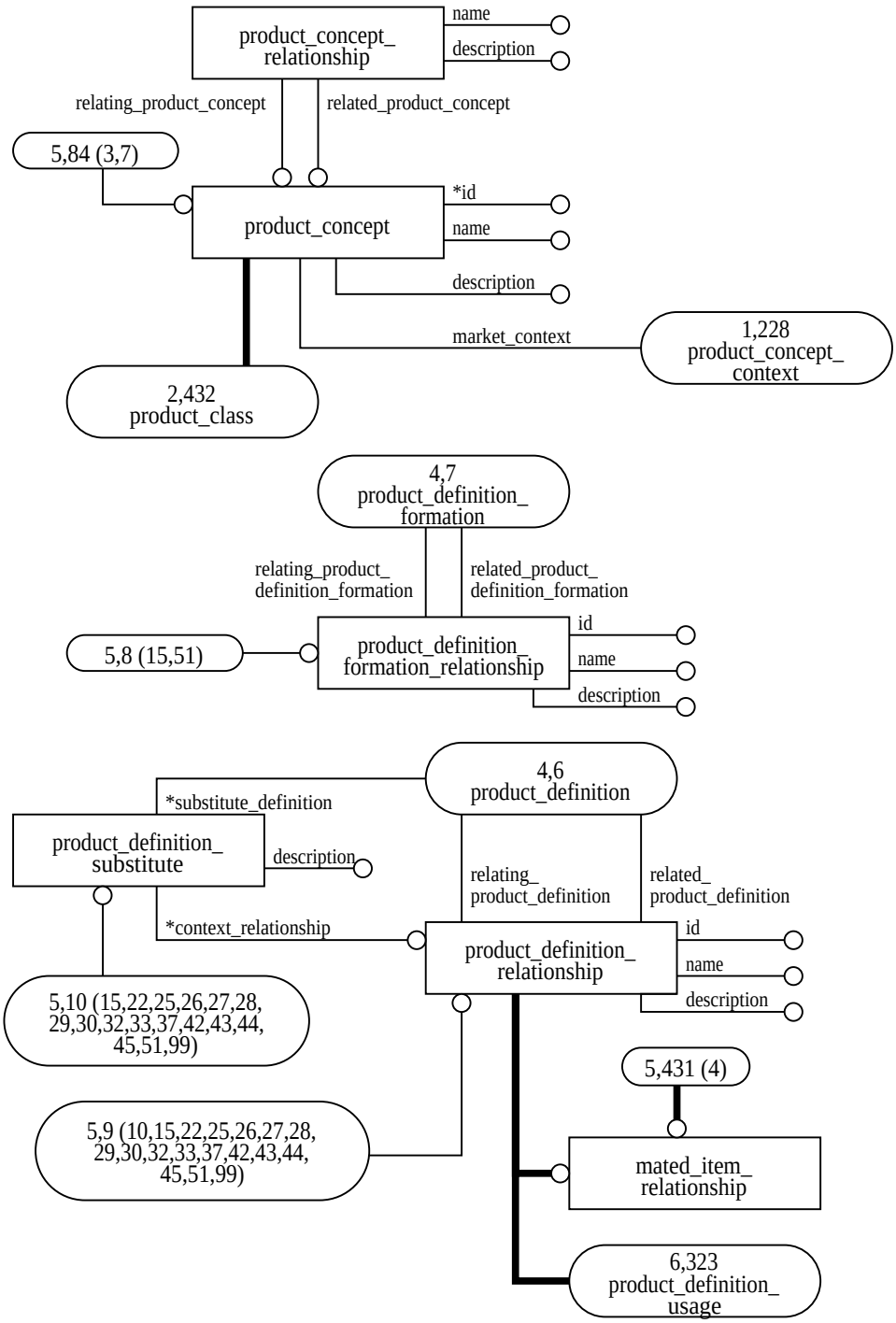


Figure H.5 – AIM diagram in EXPRESS_G: 5 of 124

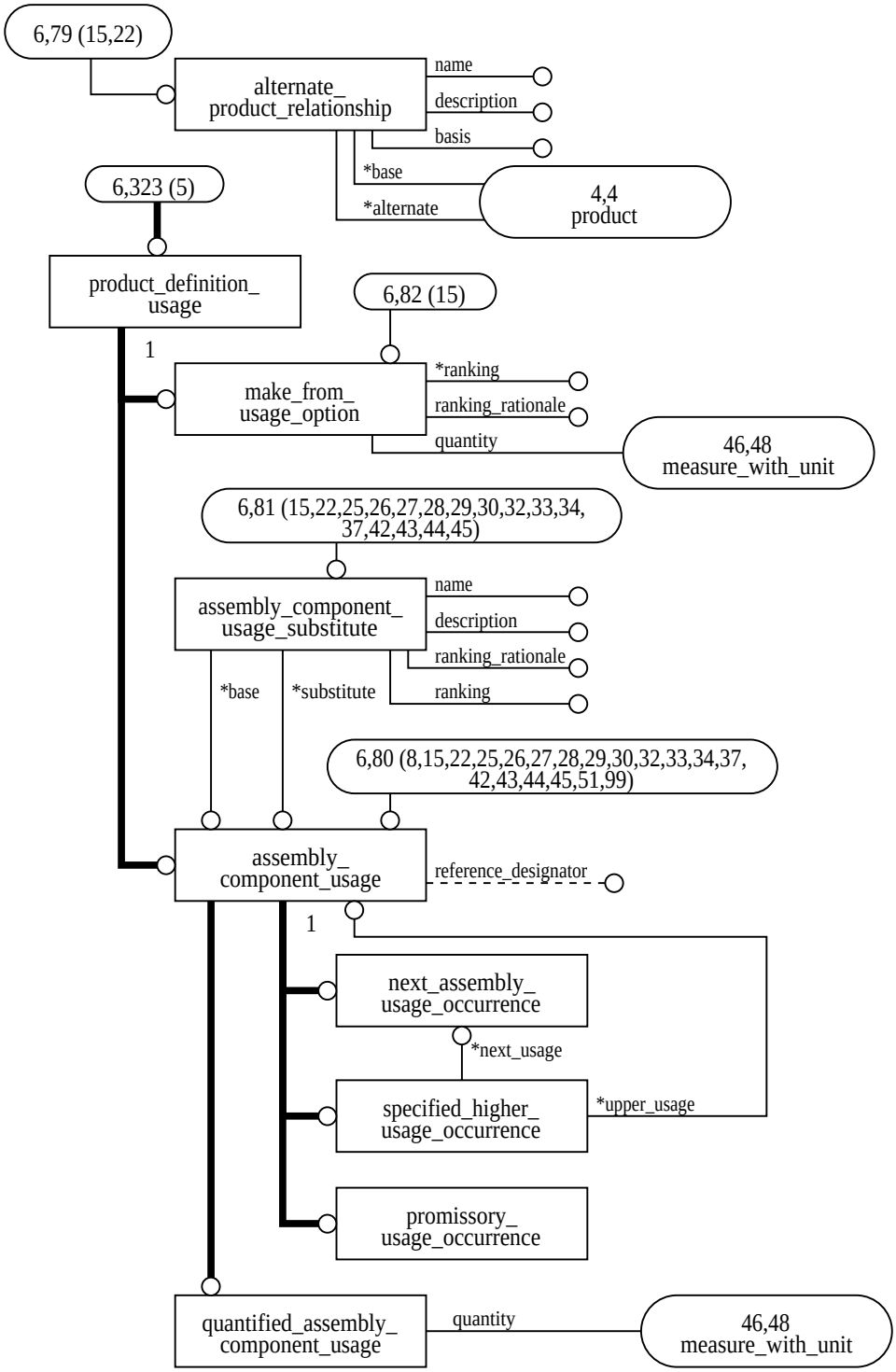


Figure H.6 – AIM diagram in EXPRESS_G: 6 of 124

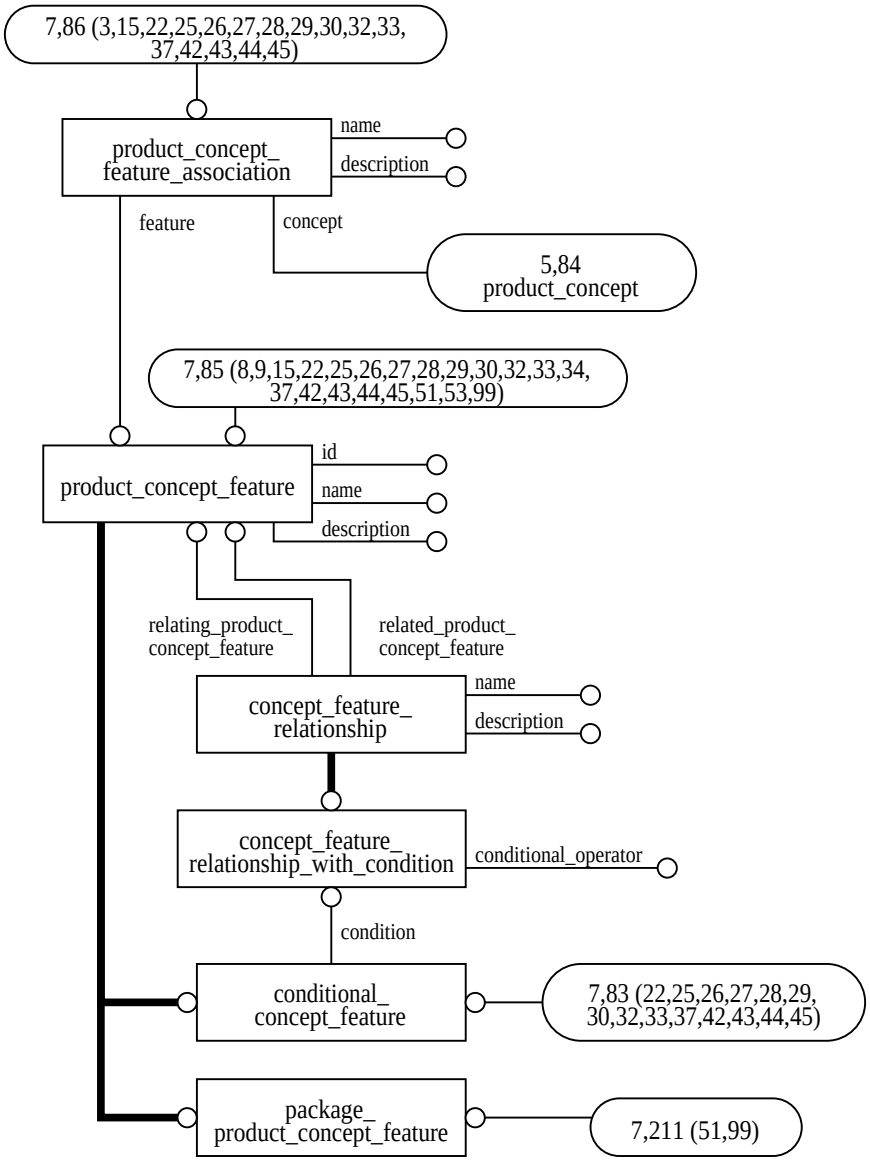


Figure H.7 – AIM diagram in EXPRESS_G: 7 of 124

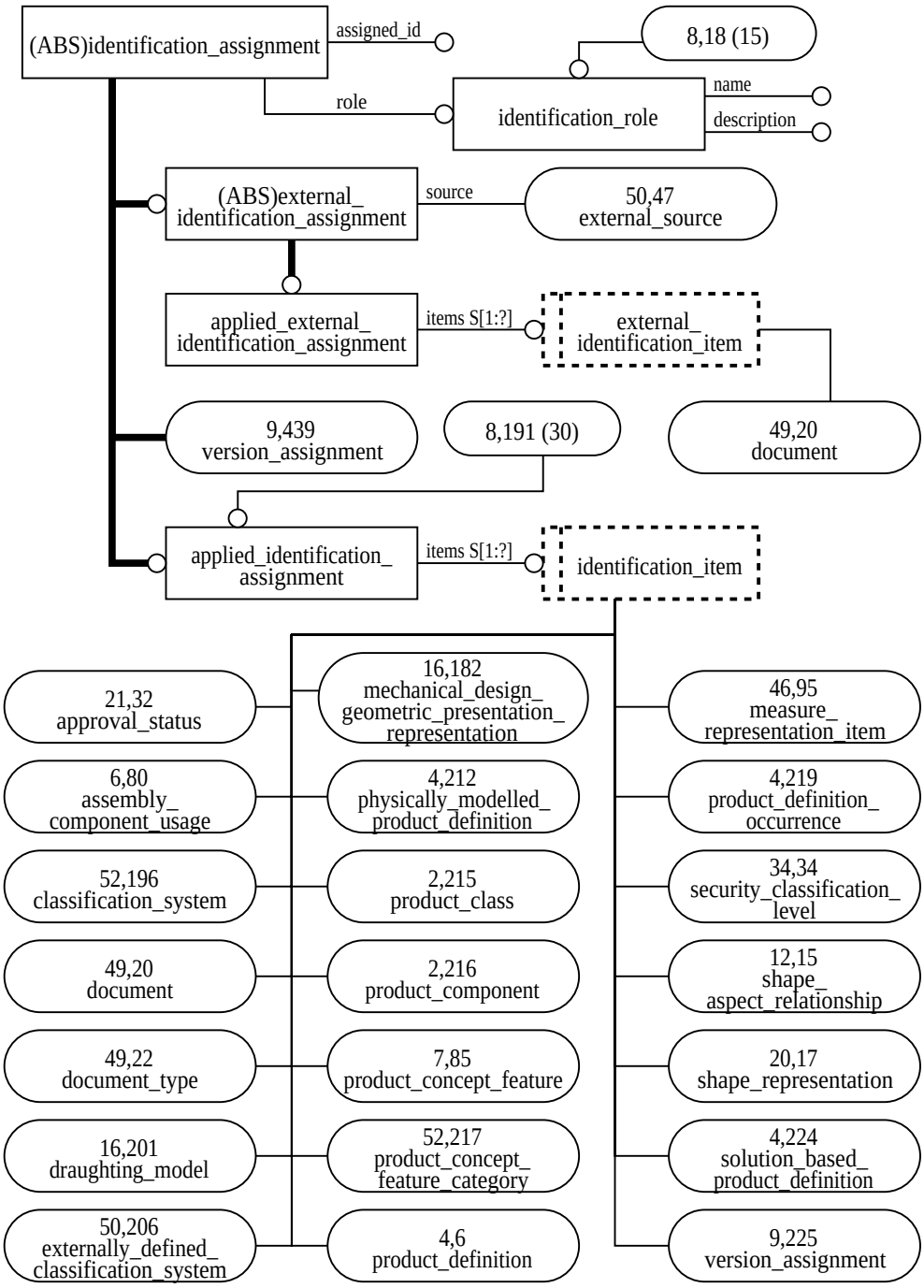


Figure H.8 – AIM diagram in EXPRESS_G: 8 of 124

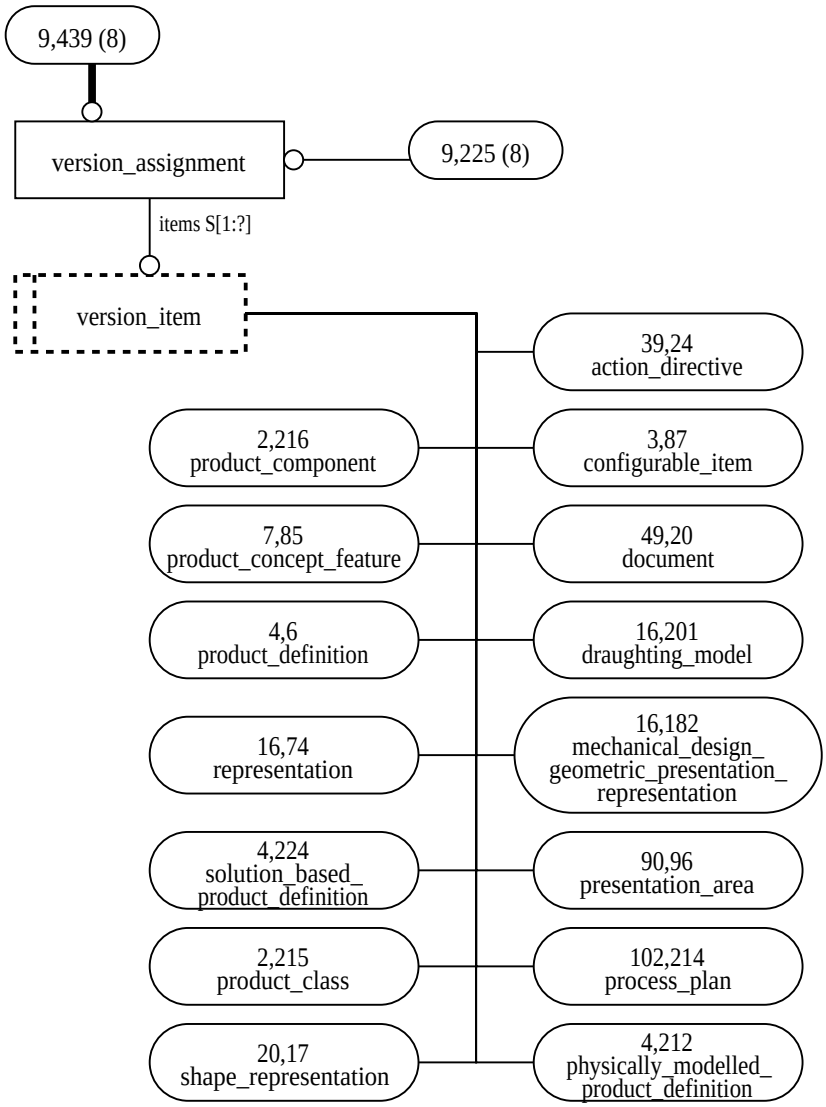


Figure H.9 – AIM diagram in EXPRESS_G: 9 of 124

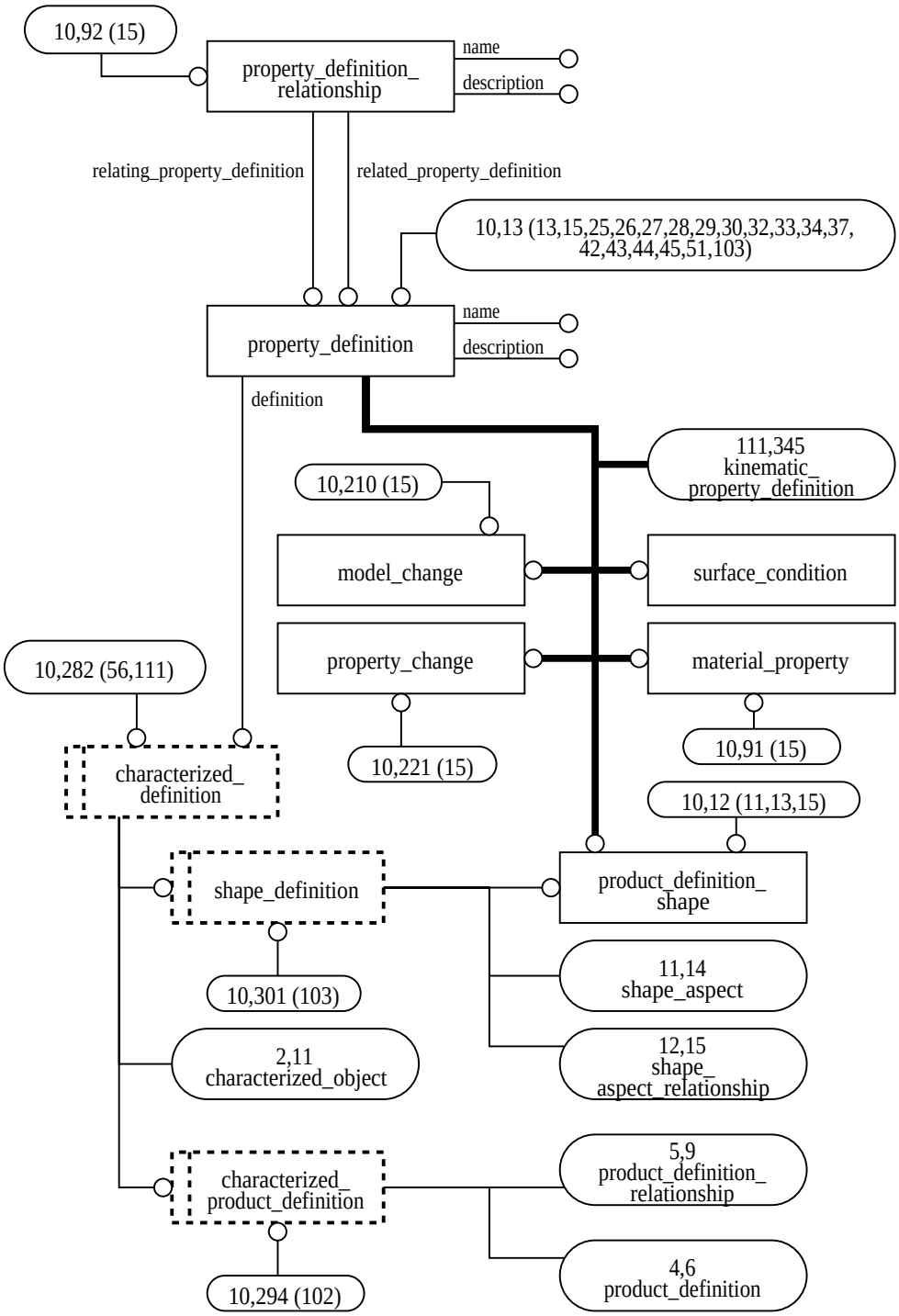


Figure H.10 – AIM diagram in EXPRESS_G: 10 of 124

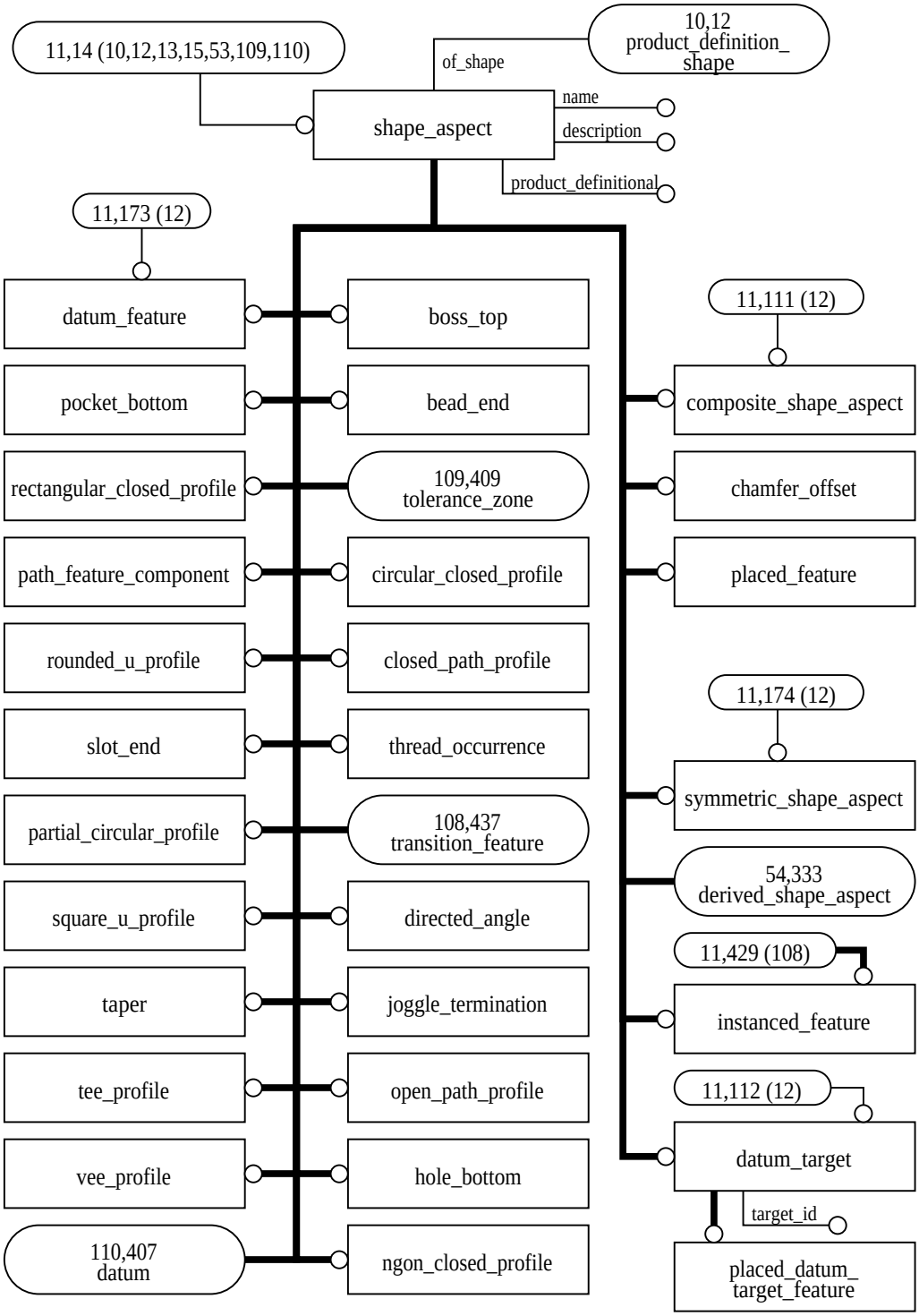


Figure H.11 – AIM diagram in EXPRESS_G: 11 of 124

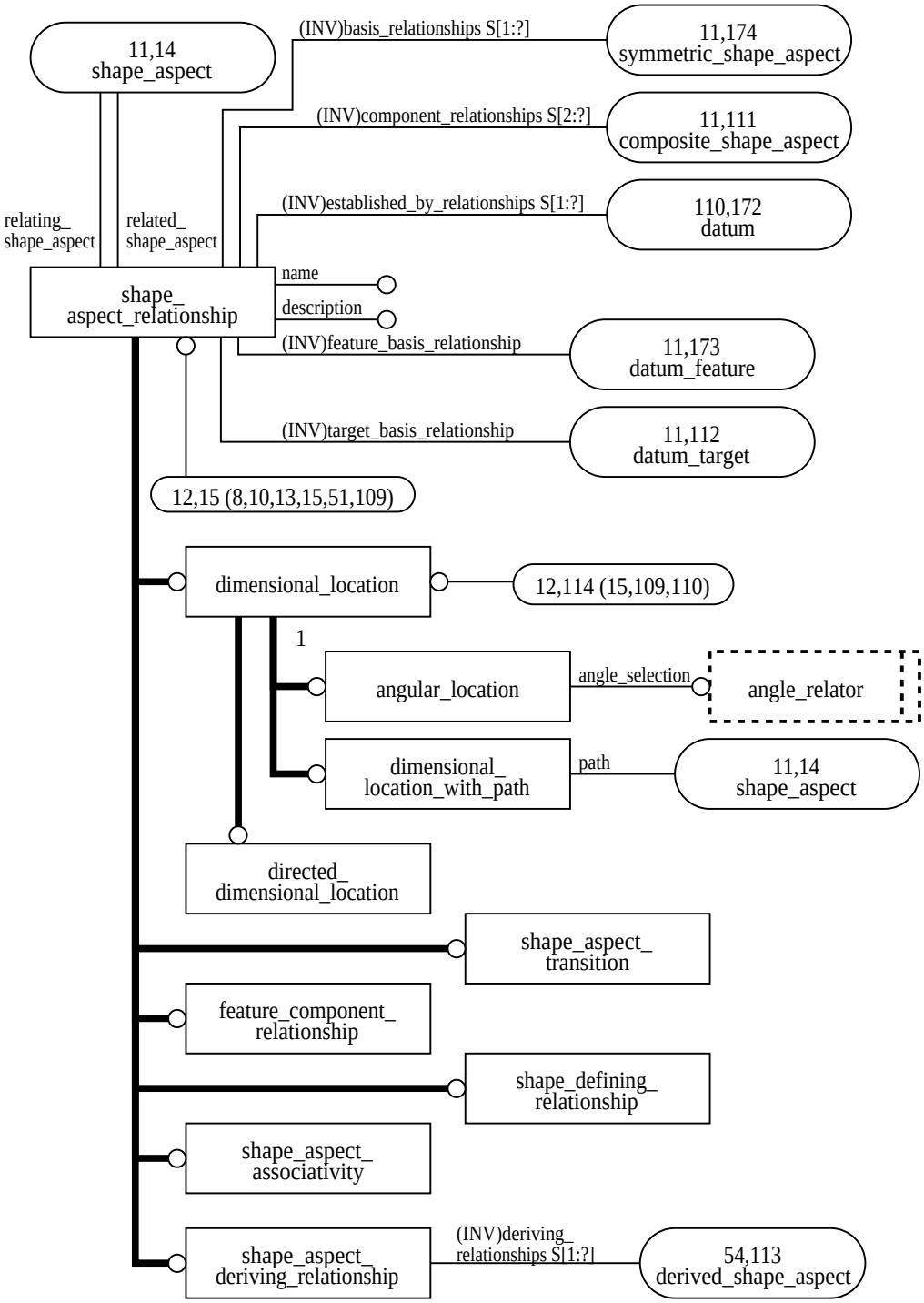


Figure H.12 – AIM diagram in EXPRESS_G: 12 of 124

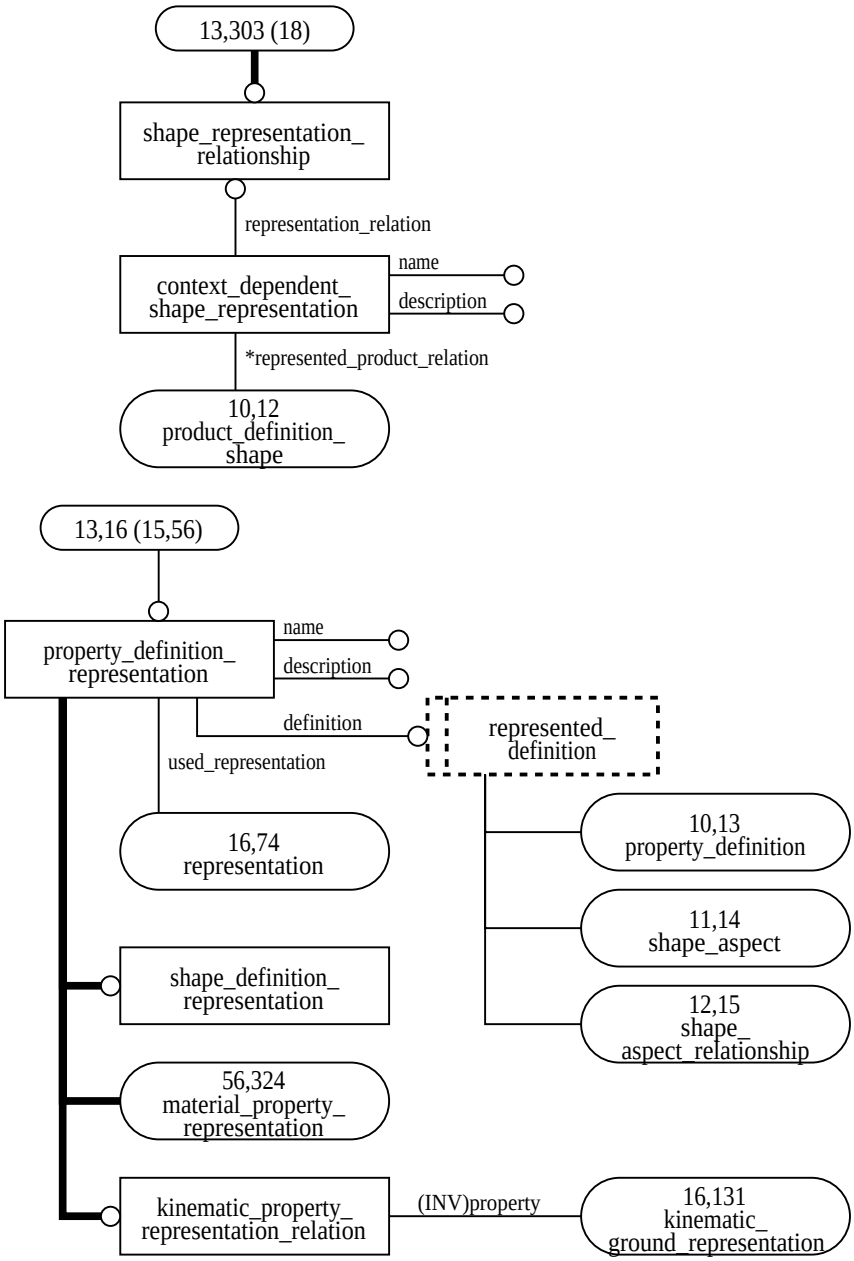


Figure H.13 – AIM diagram in EXPRESS_G: 13 of 124

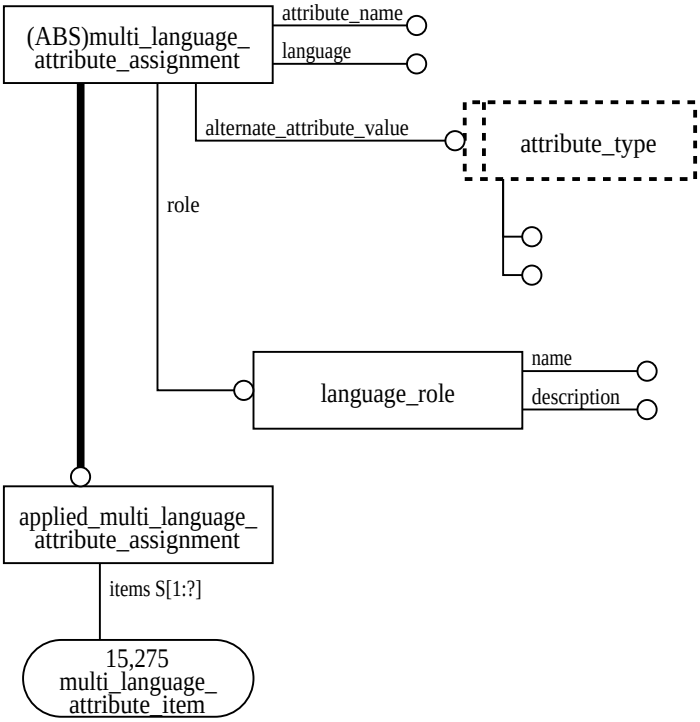


Figure H.14 – AIM diagram in EXPRESS_G: 14 of 124

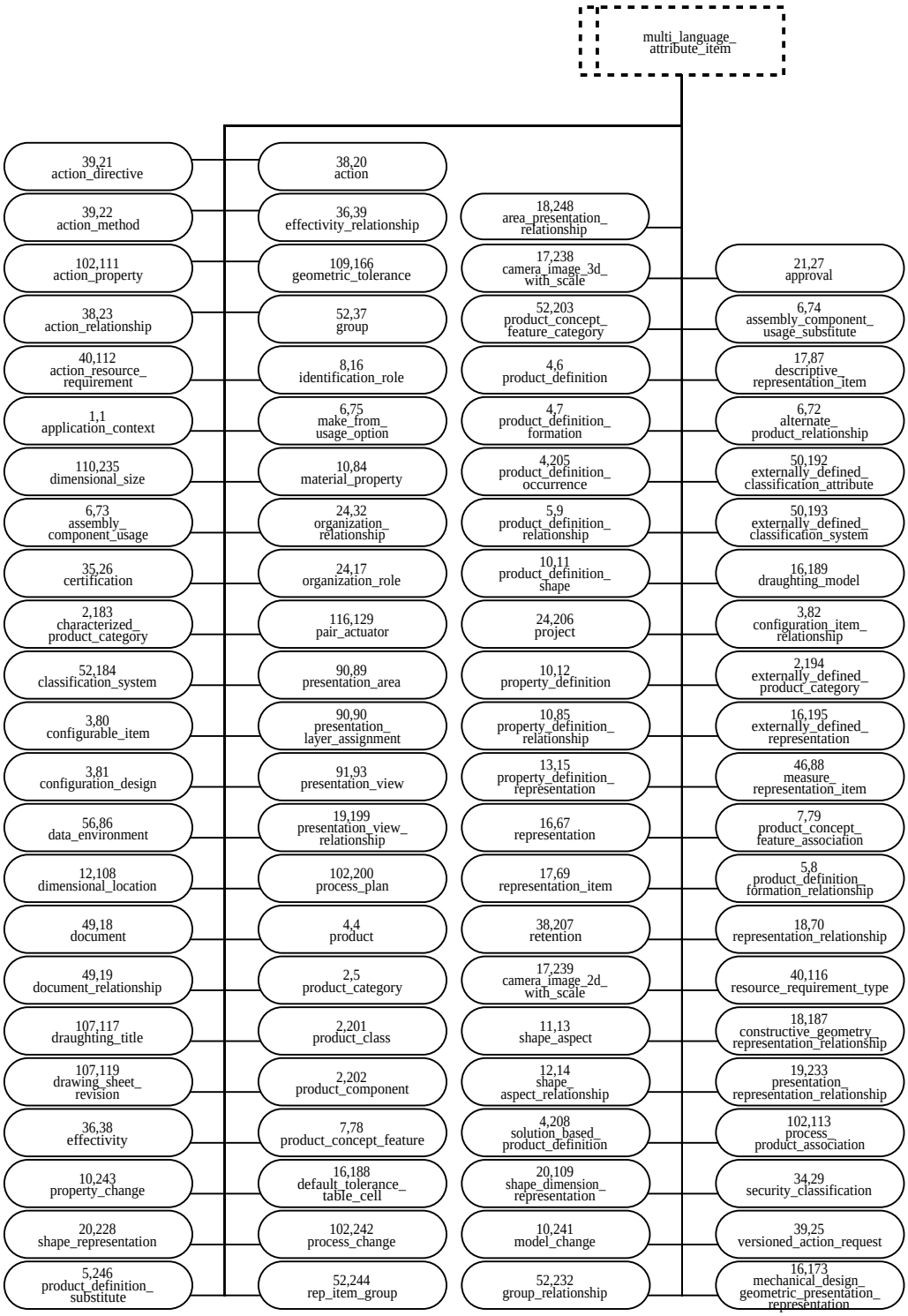


Figure H.15 – AIM diagram in EXPRESS_G: 15 of 124

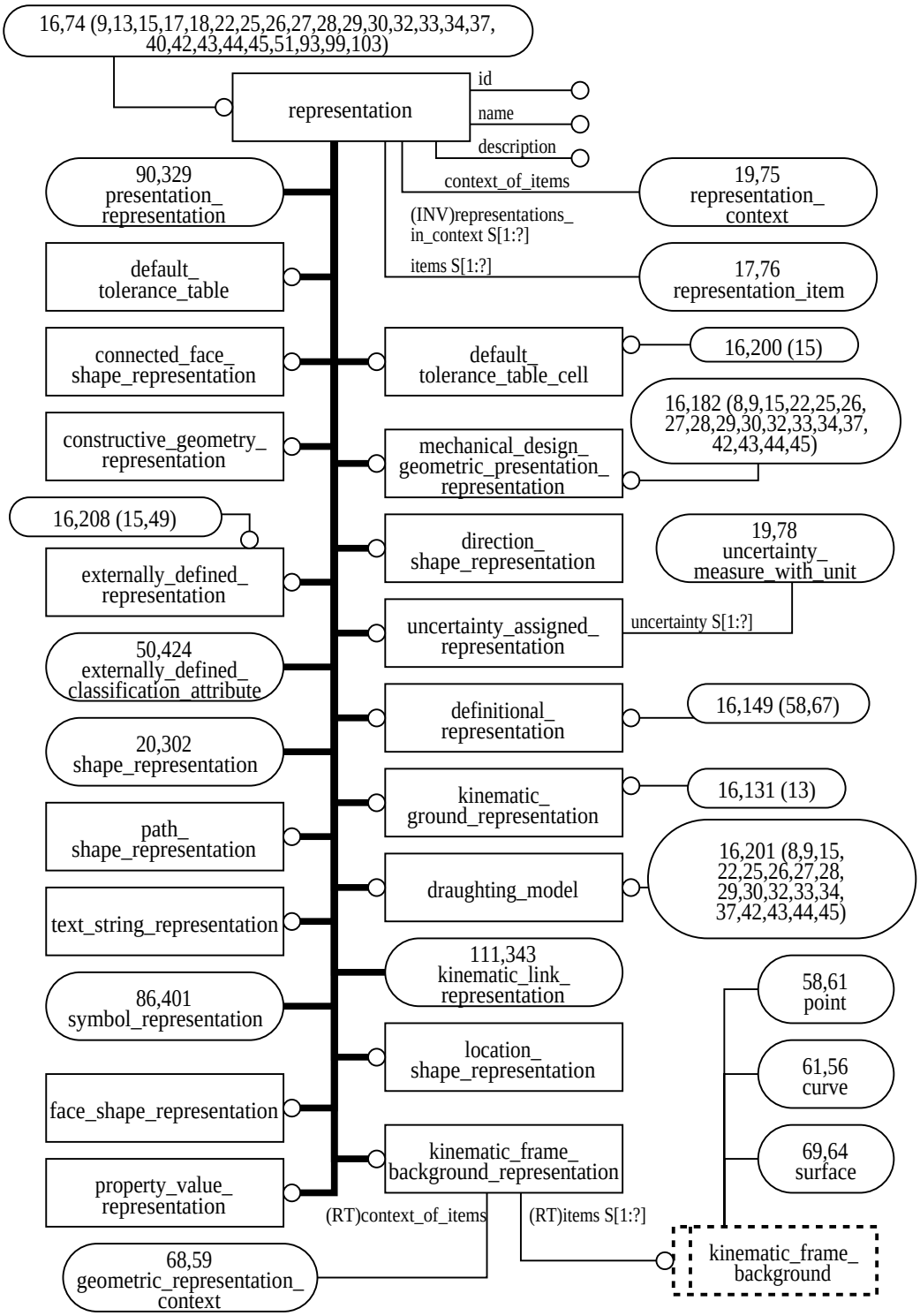


Figure H.16 – AIM diagram in EXPRESS_G: 16 of 124

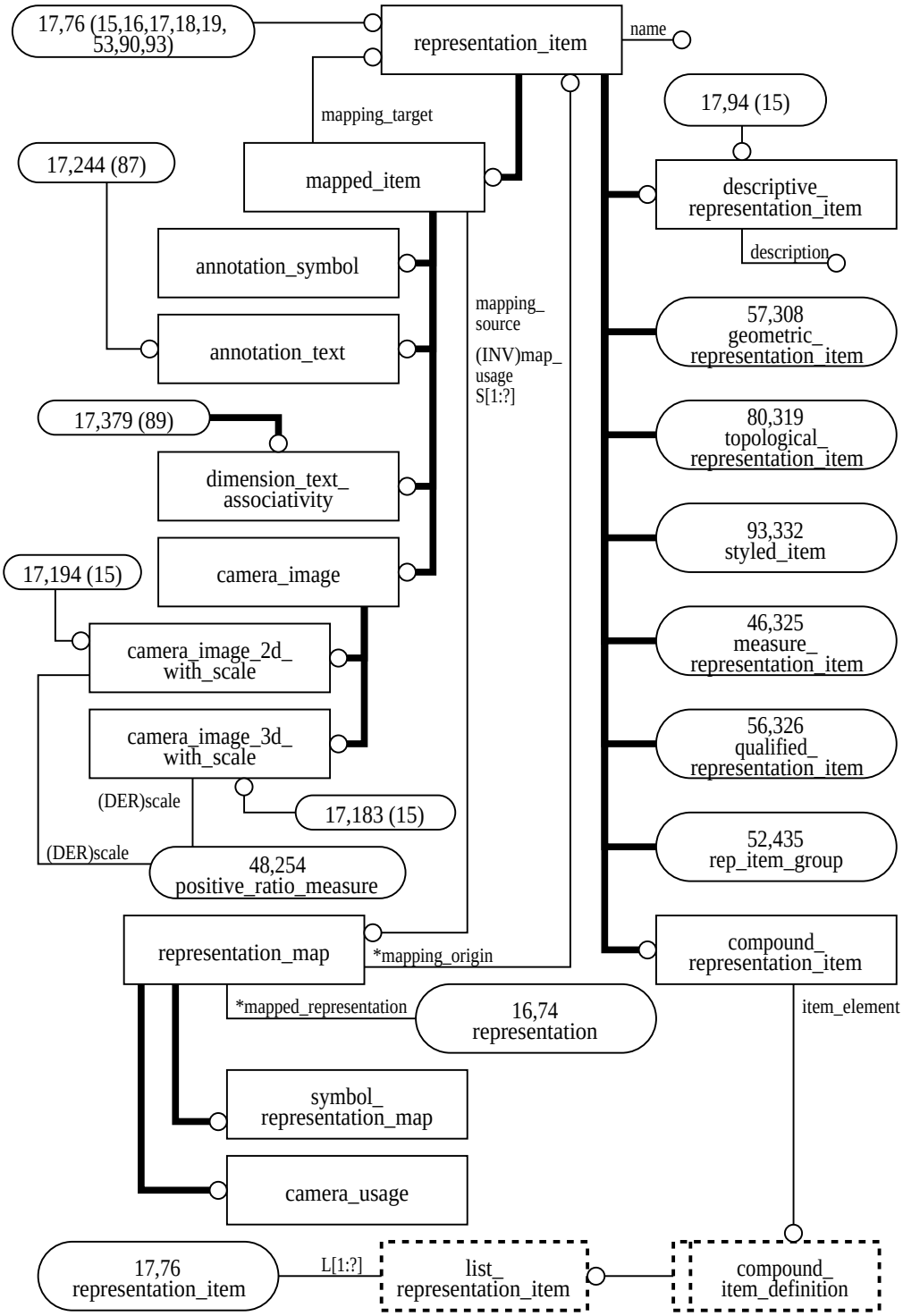


Figure H.17 – AIM diagram in EXPRESS_G: 17 of 124

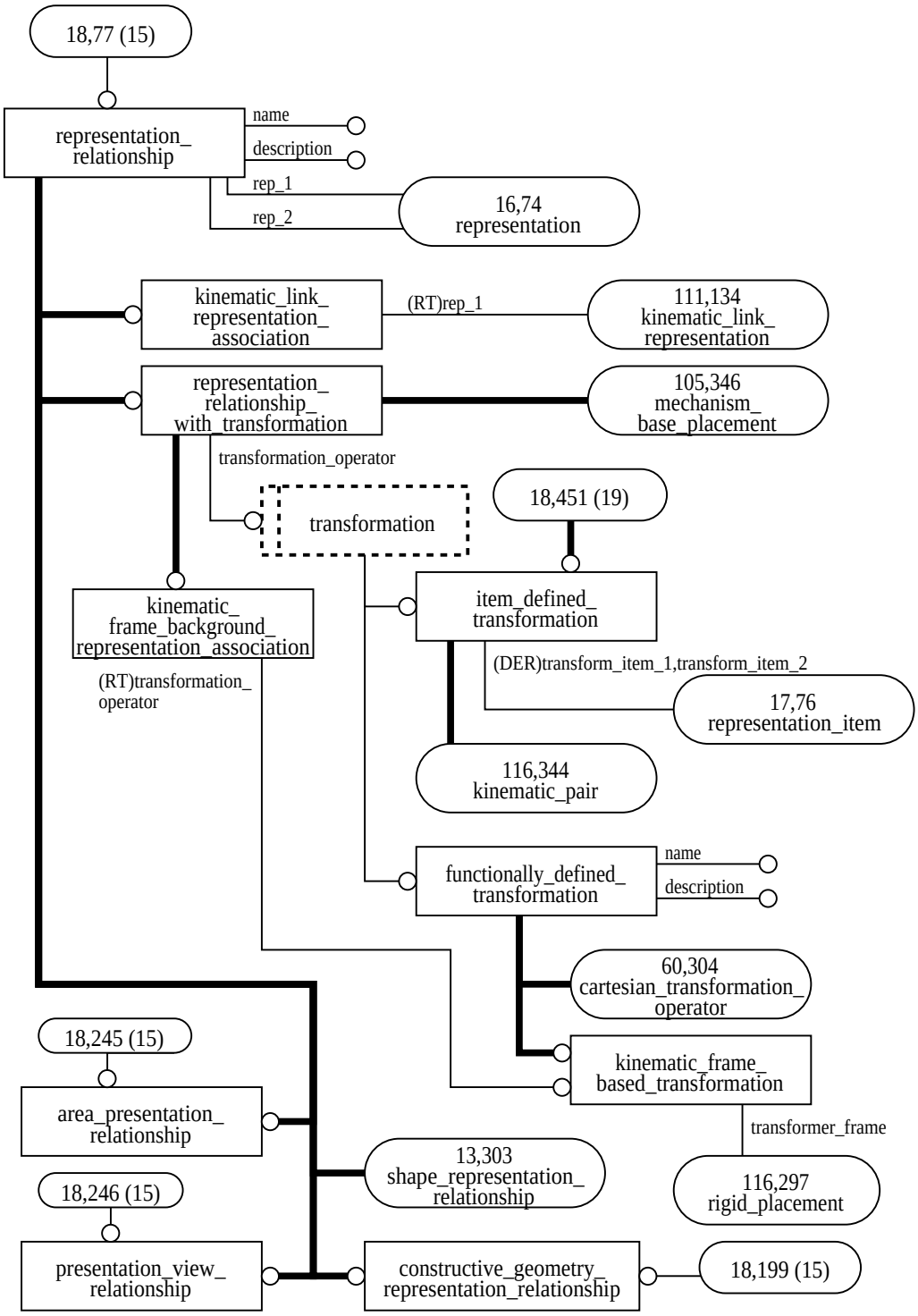


Figure H.18 – AIM diagram in EXPRESS_G: 18 of 124

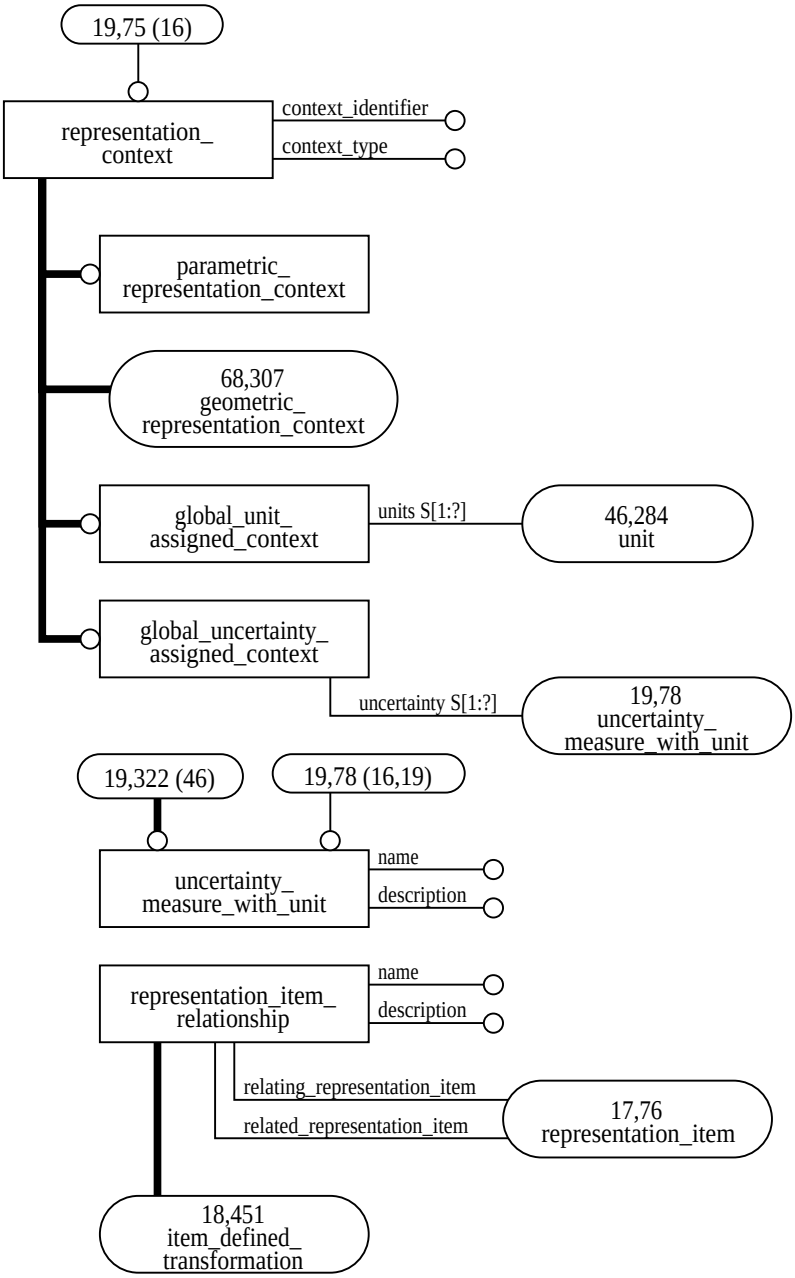


Figure H.19 – AIM diagram in EXPRESS_G: 19 of 124

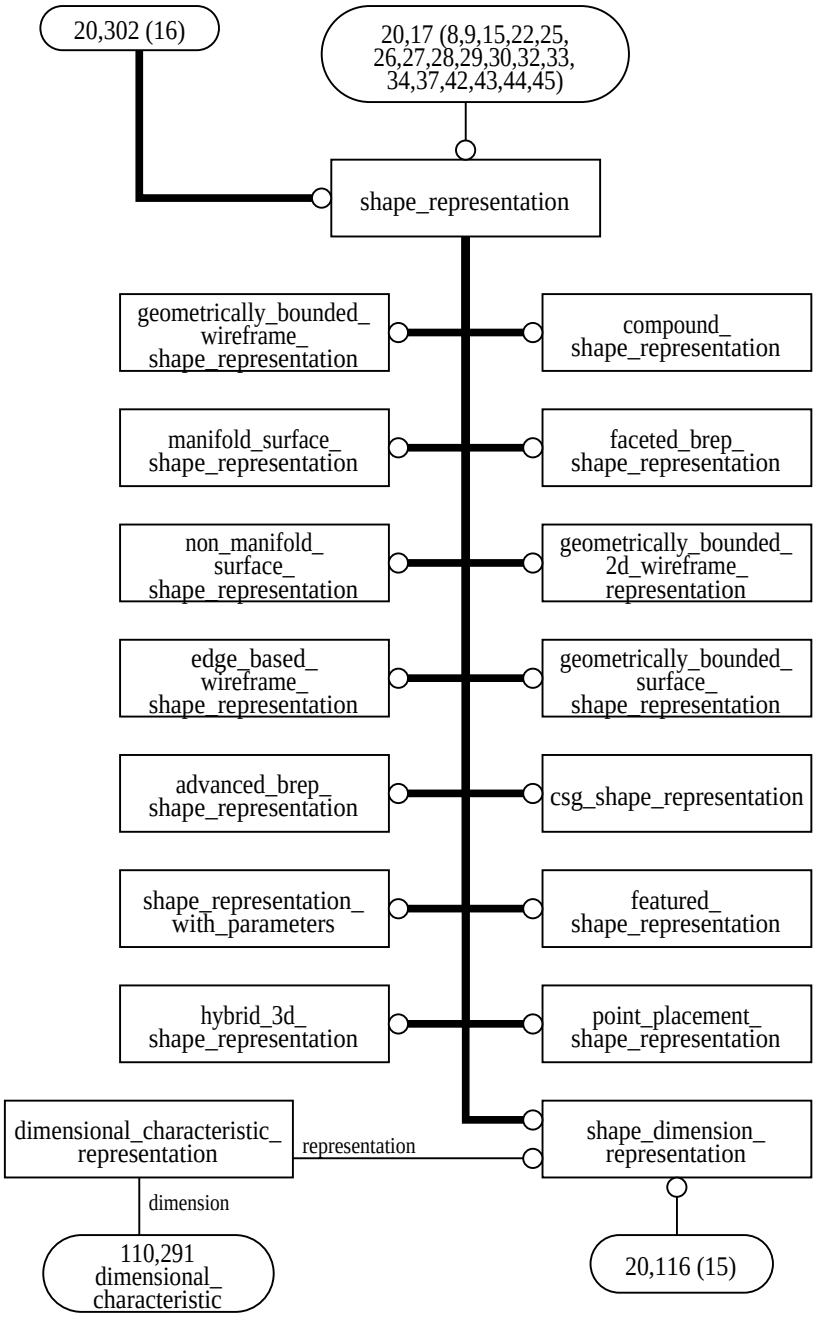


Figure H.20 – AIM diagram in EXPRESS_G: 20 of 124

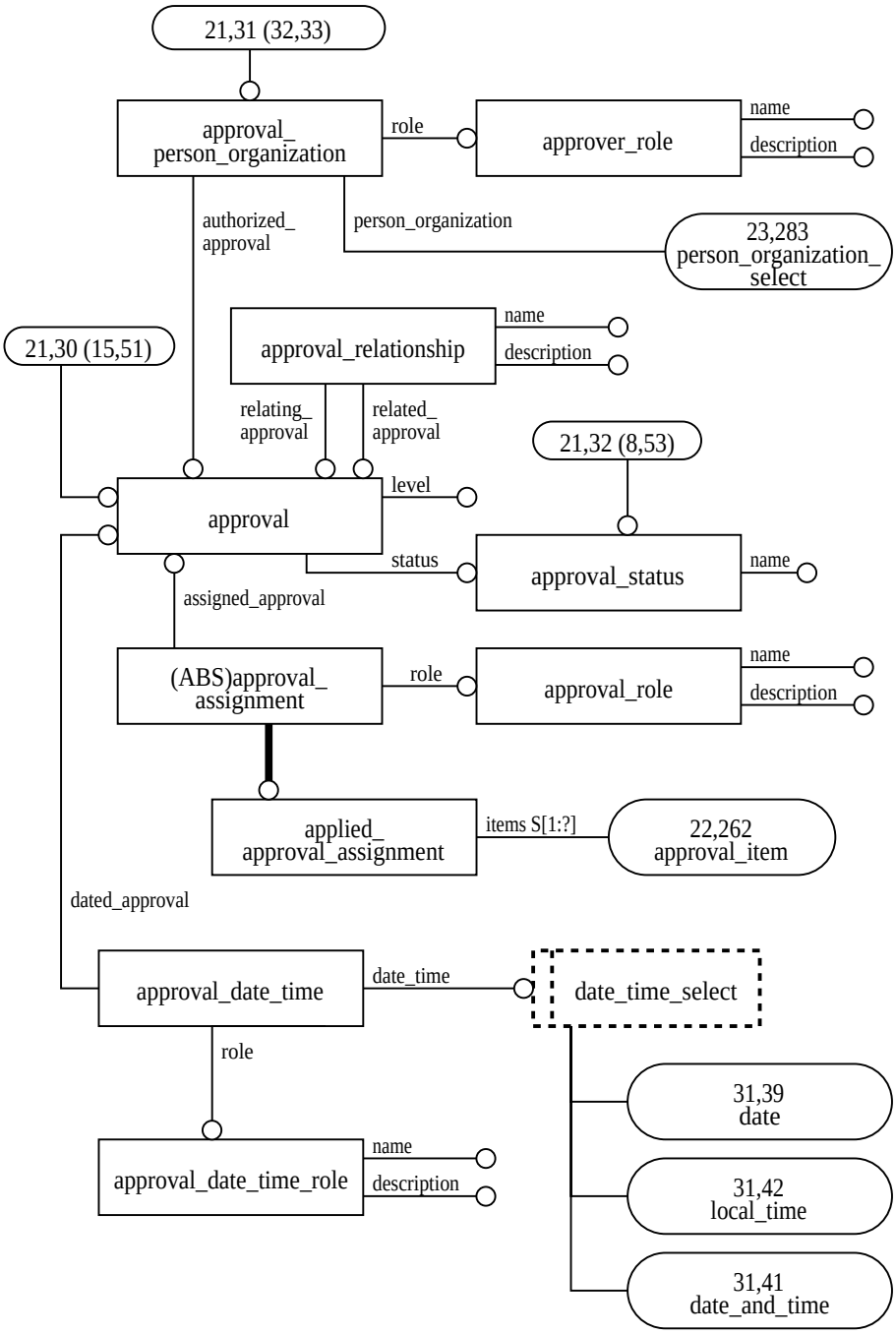


Figure H.21 – AIM diagram in EXPRESS_G: 21 of 124

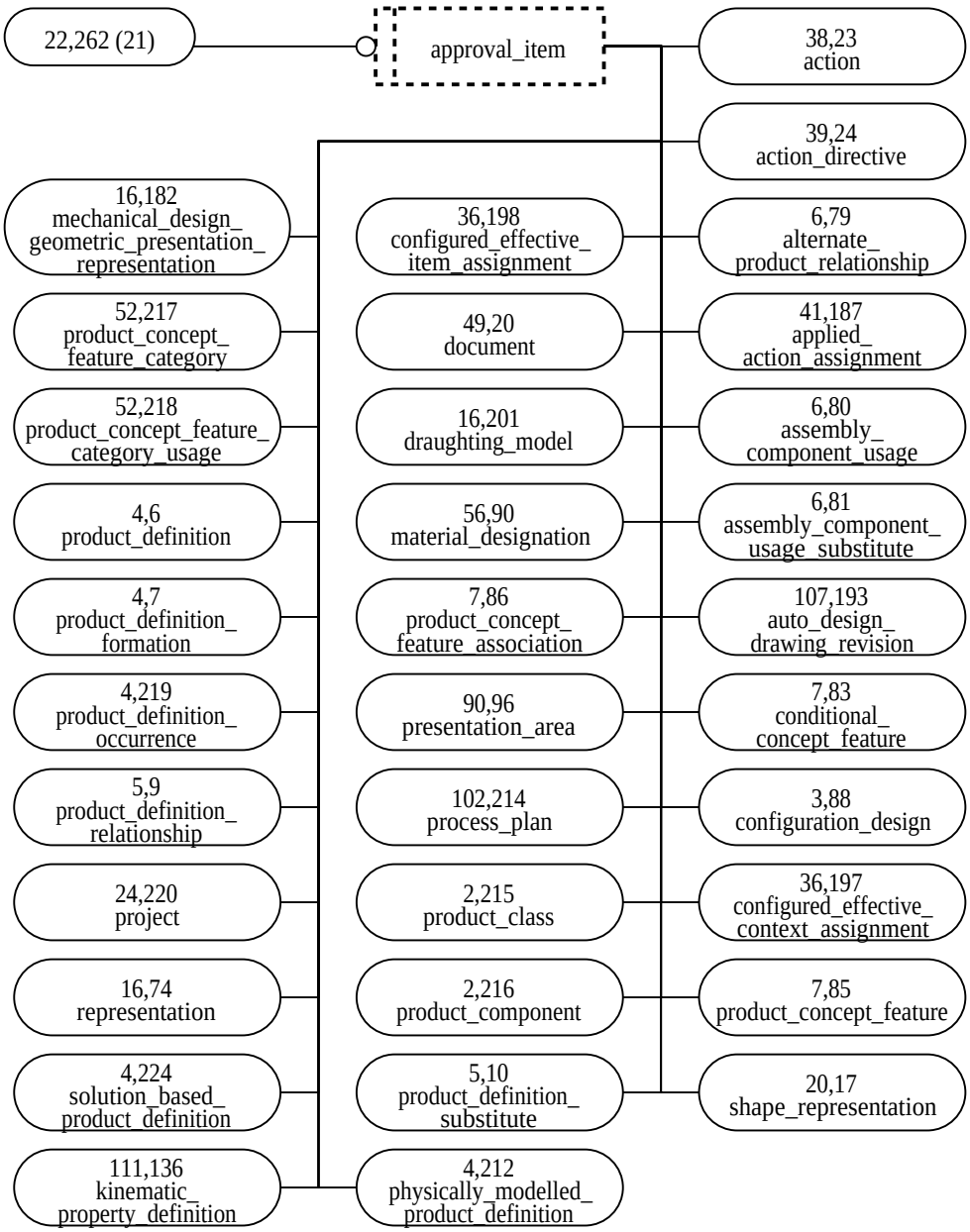


Figure H.22 – AIM diagram in EXPRESS_G: 22 of 124

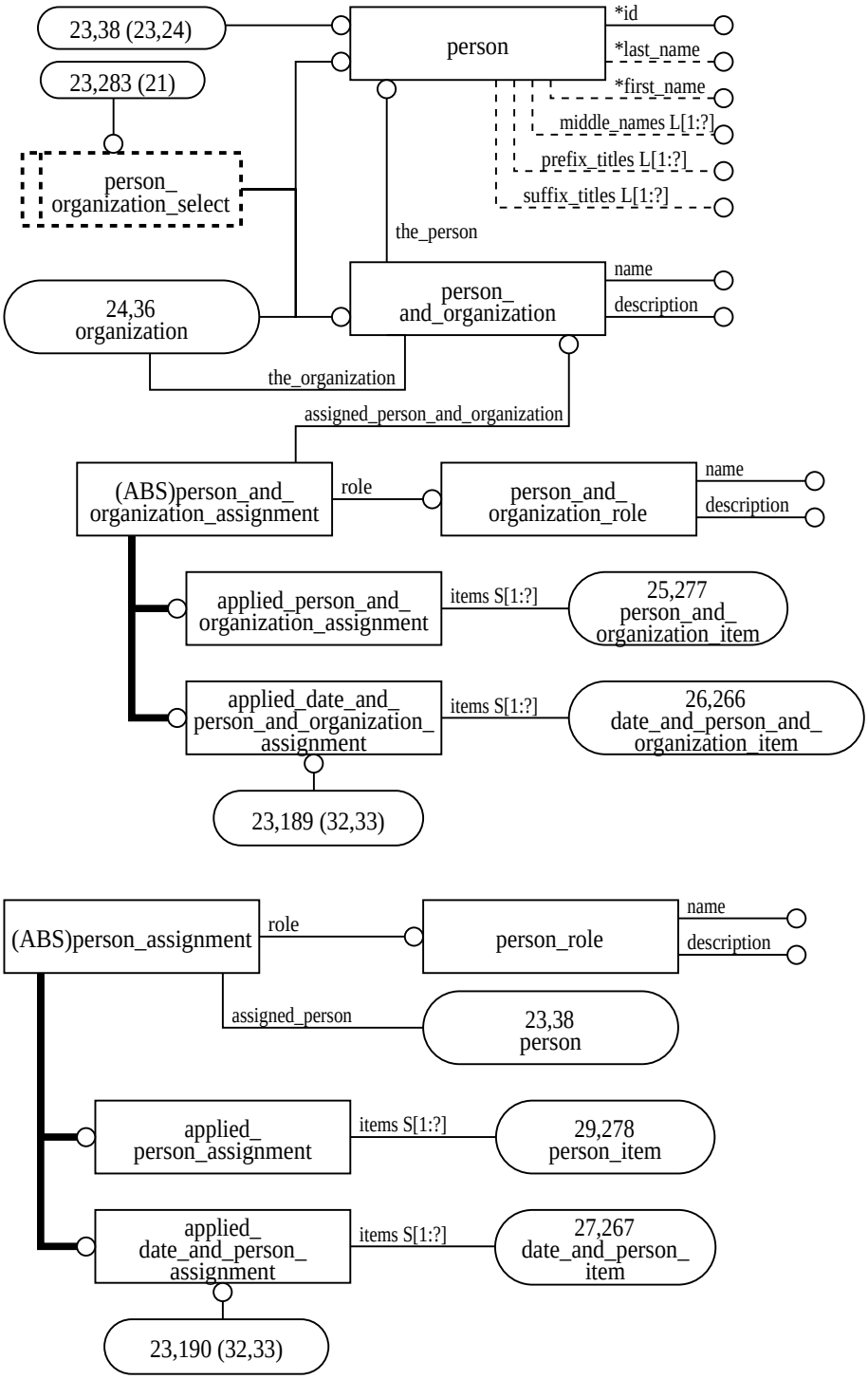


Figure H.23 – AIM diagram in EXPRESS_G: 23 of 124

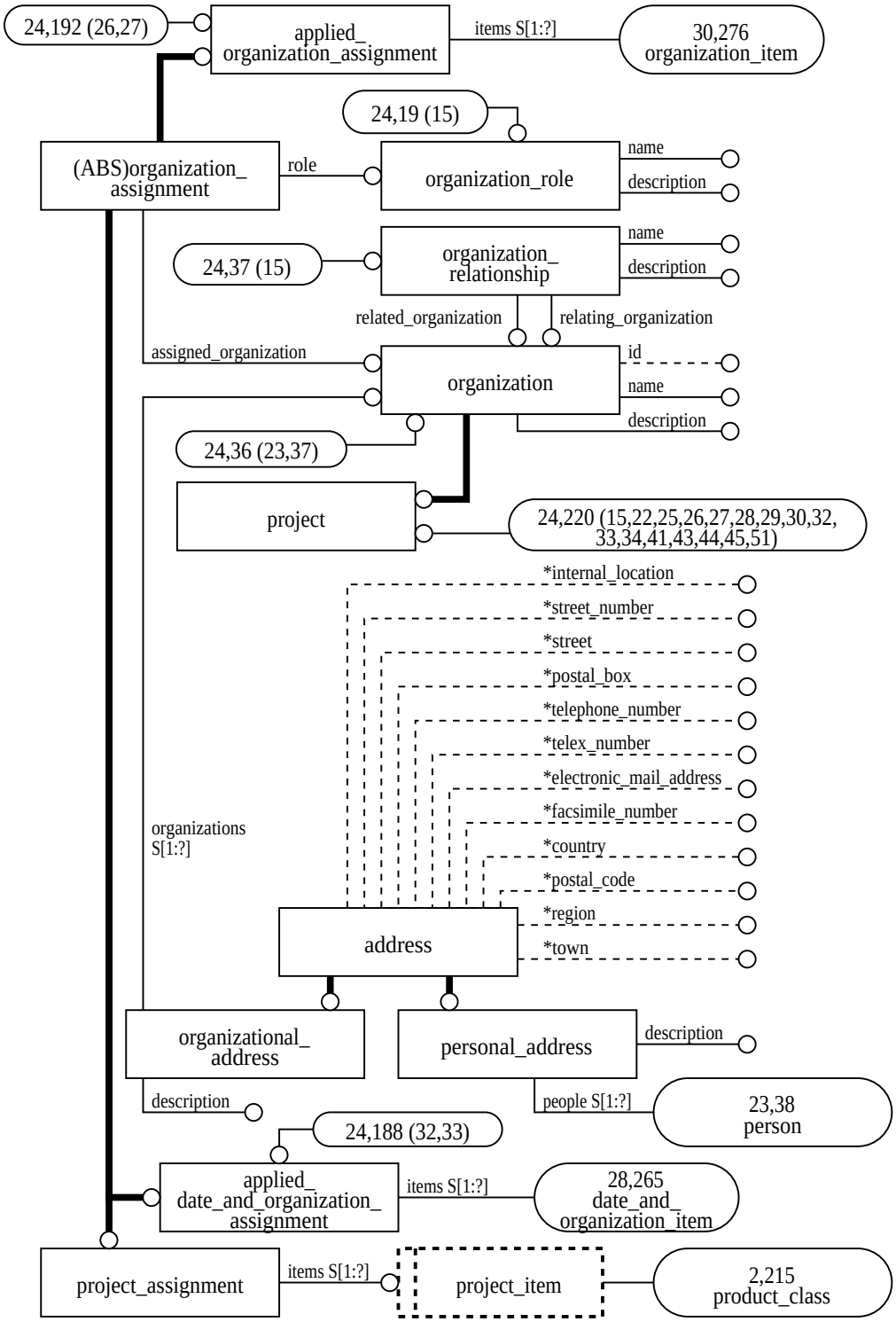


Figure H.24 – AIM diagram in EXPRESS_G: 24 of 124

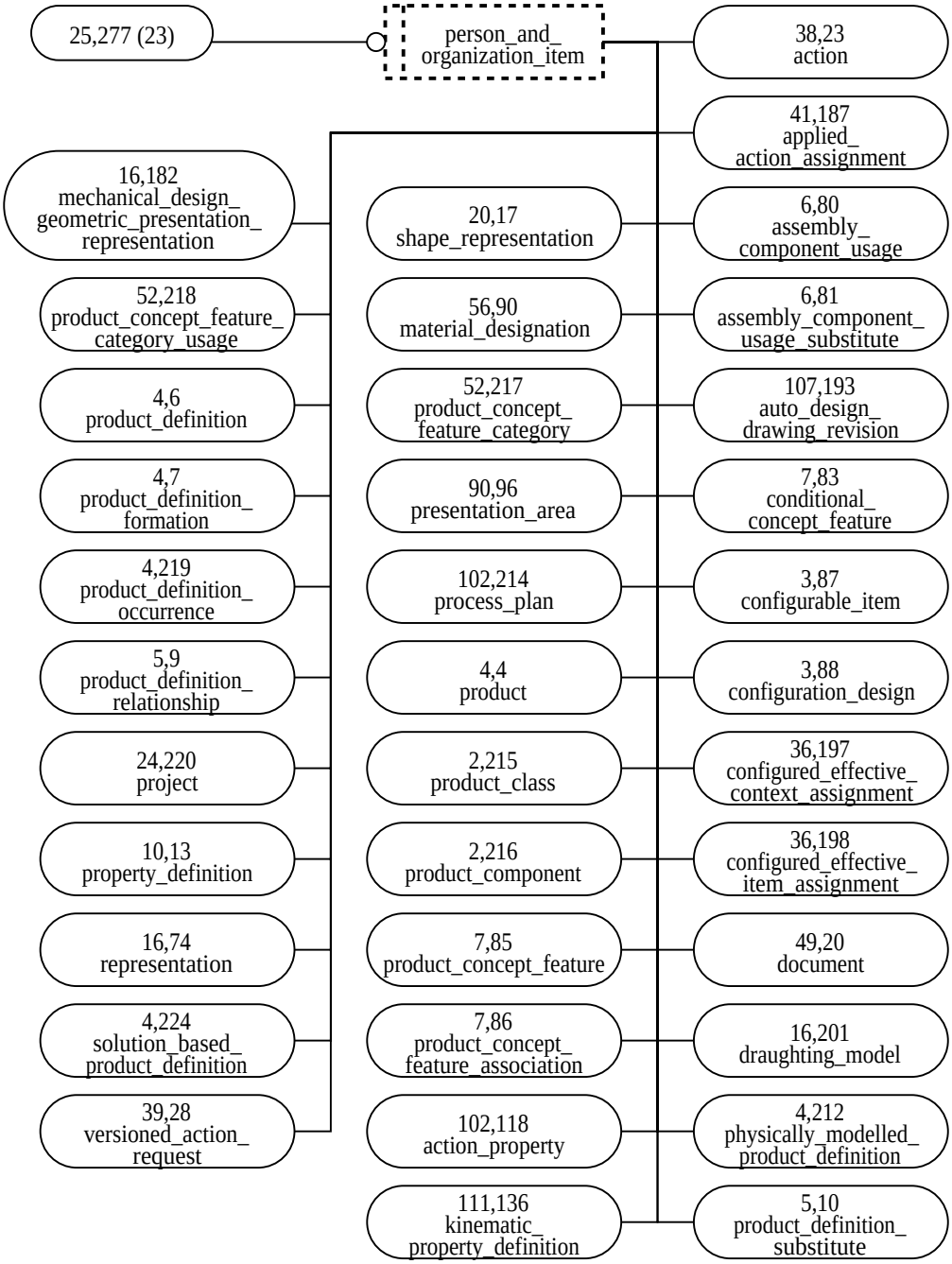


Figure H.25 – AIM diagram in EXPRESS_G: 25 of 124

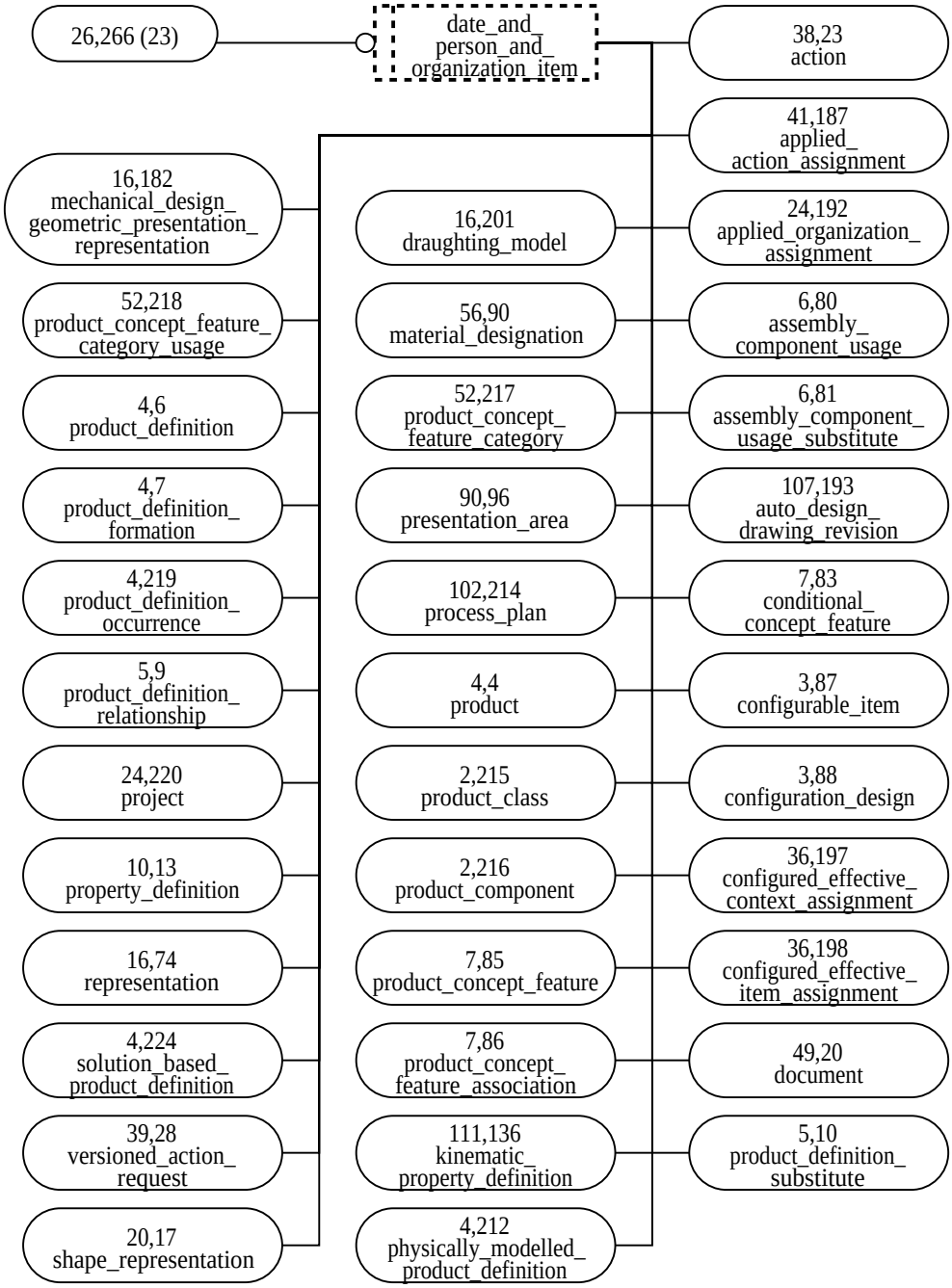


Figure H.26 – AIM diagram in EXPRESS_G: 26 of 124

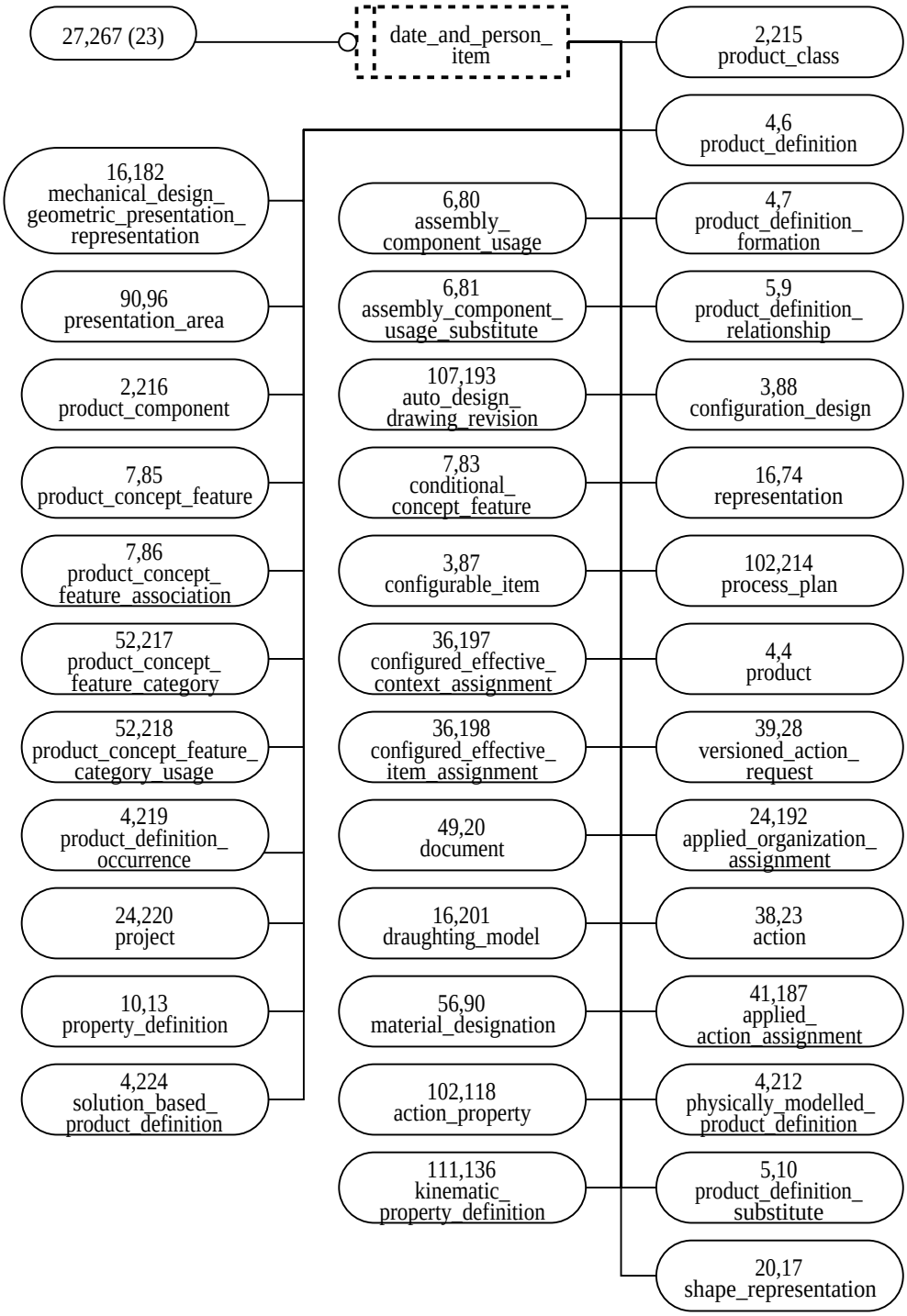


Figure H.27 – AIM diagram in EXPRESS_G: 27 of 124

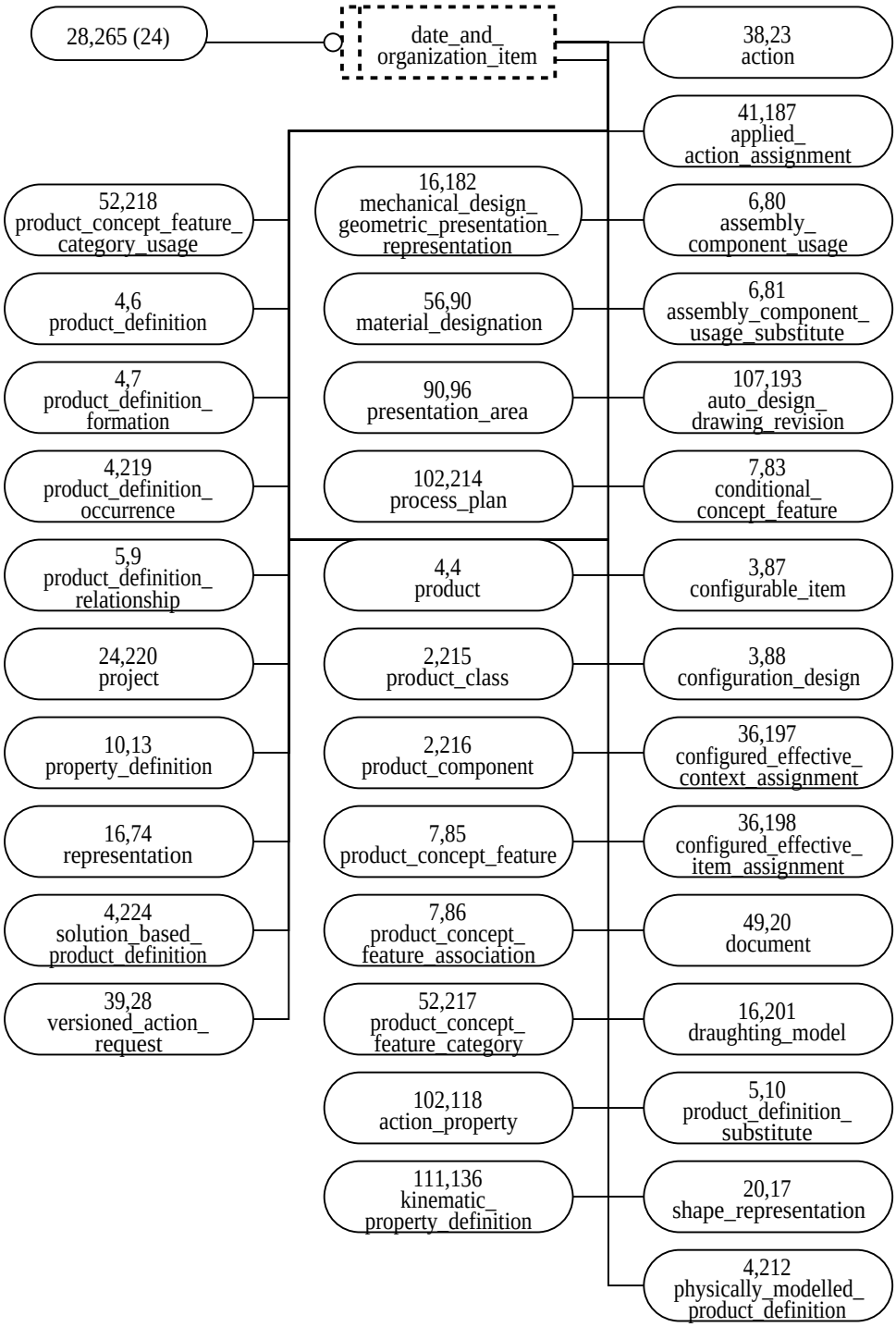


Figure H.28 – AIM diagram in EXPRESS_G: 28 of 124

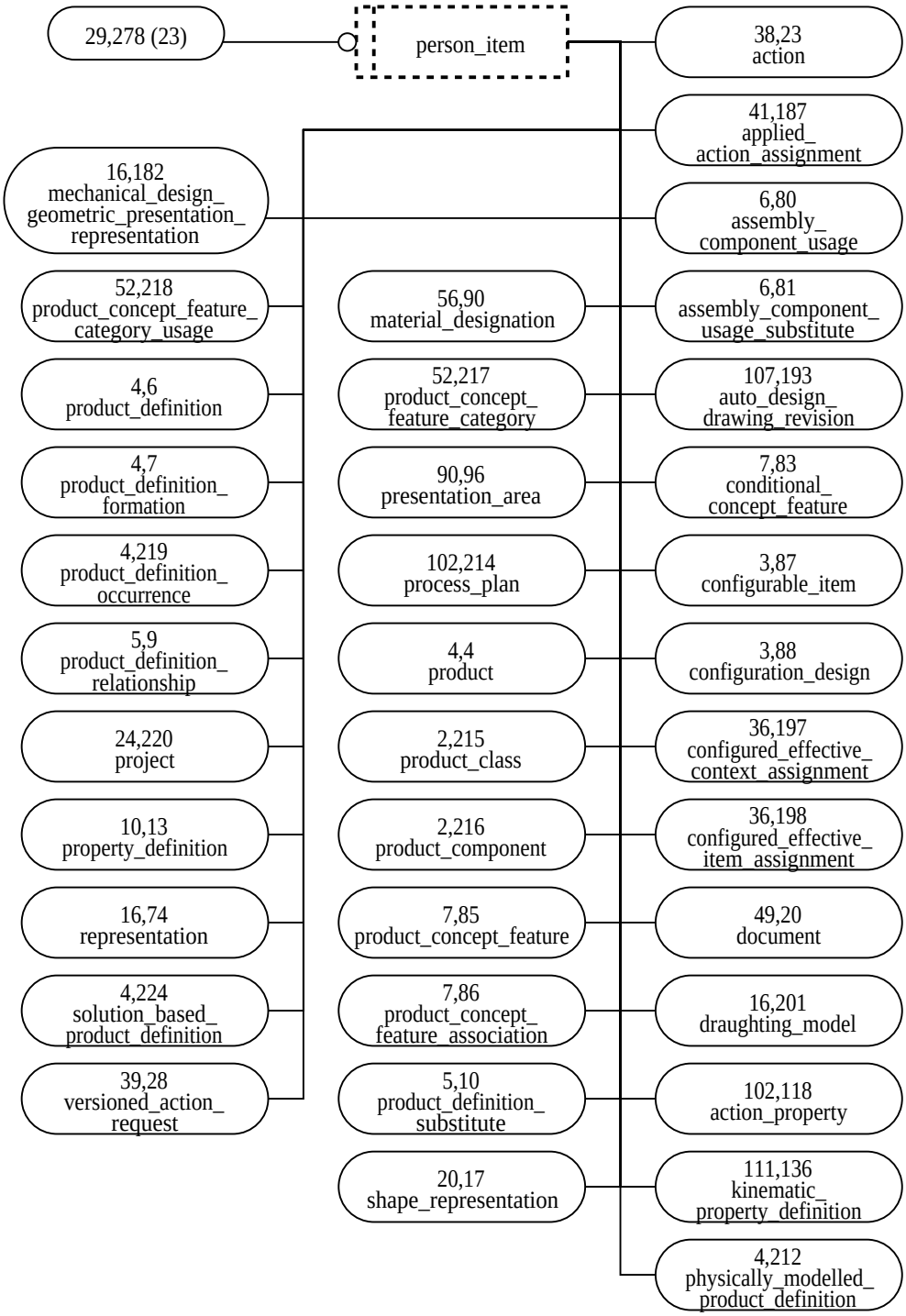


Figure H.29 – AIM diagram in EXPRESS_G: 29 of 124

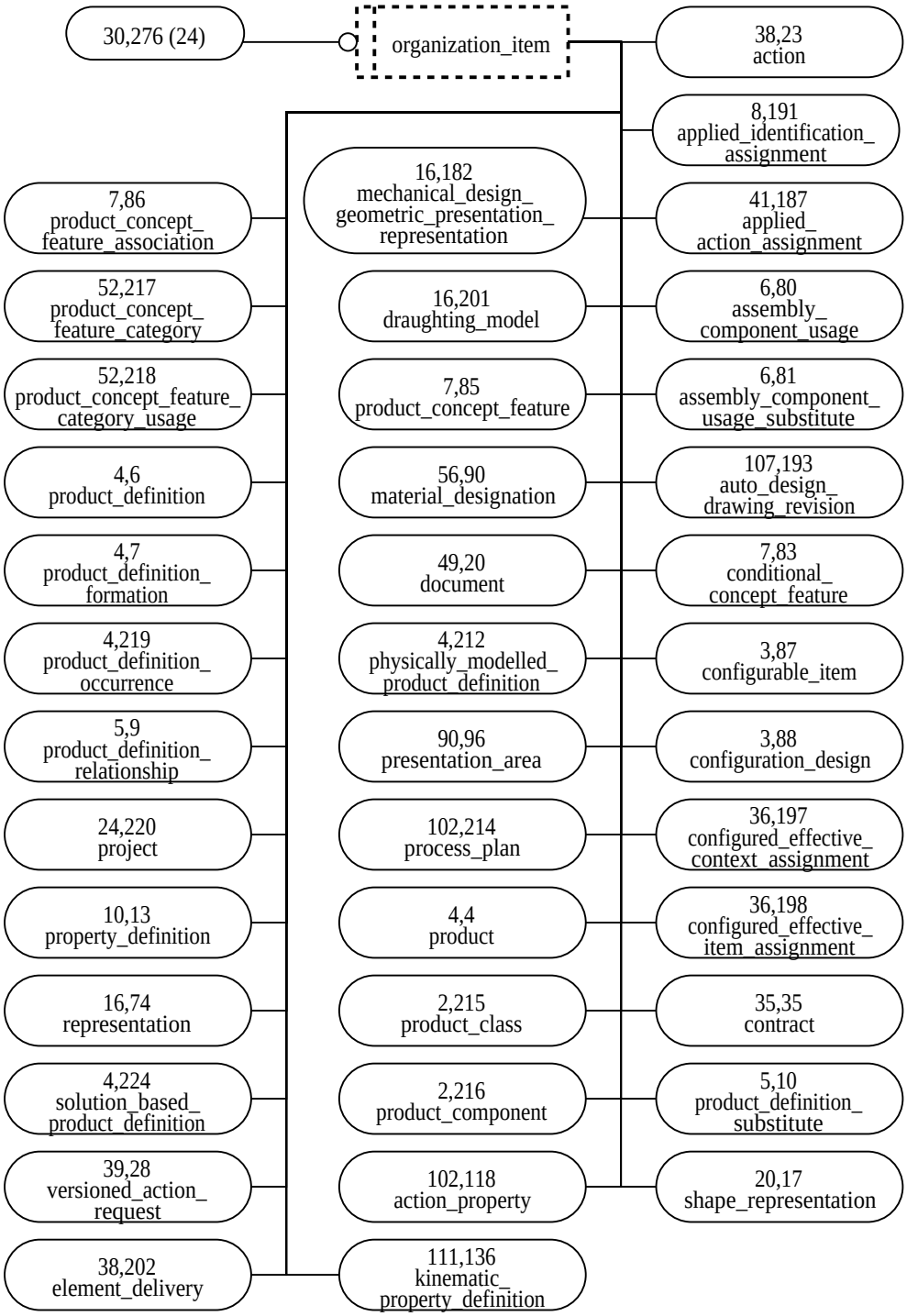


Figure H.30 – AIM diagram in EXPRESS_G: 30 of 124

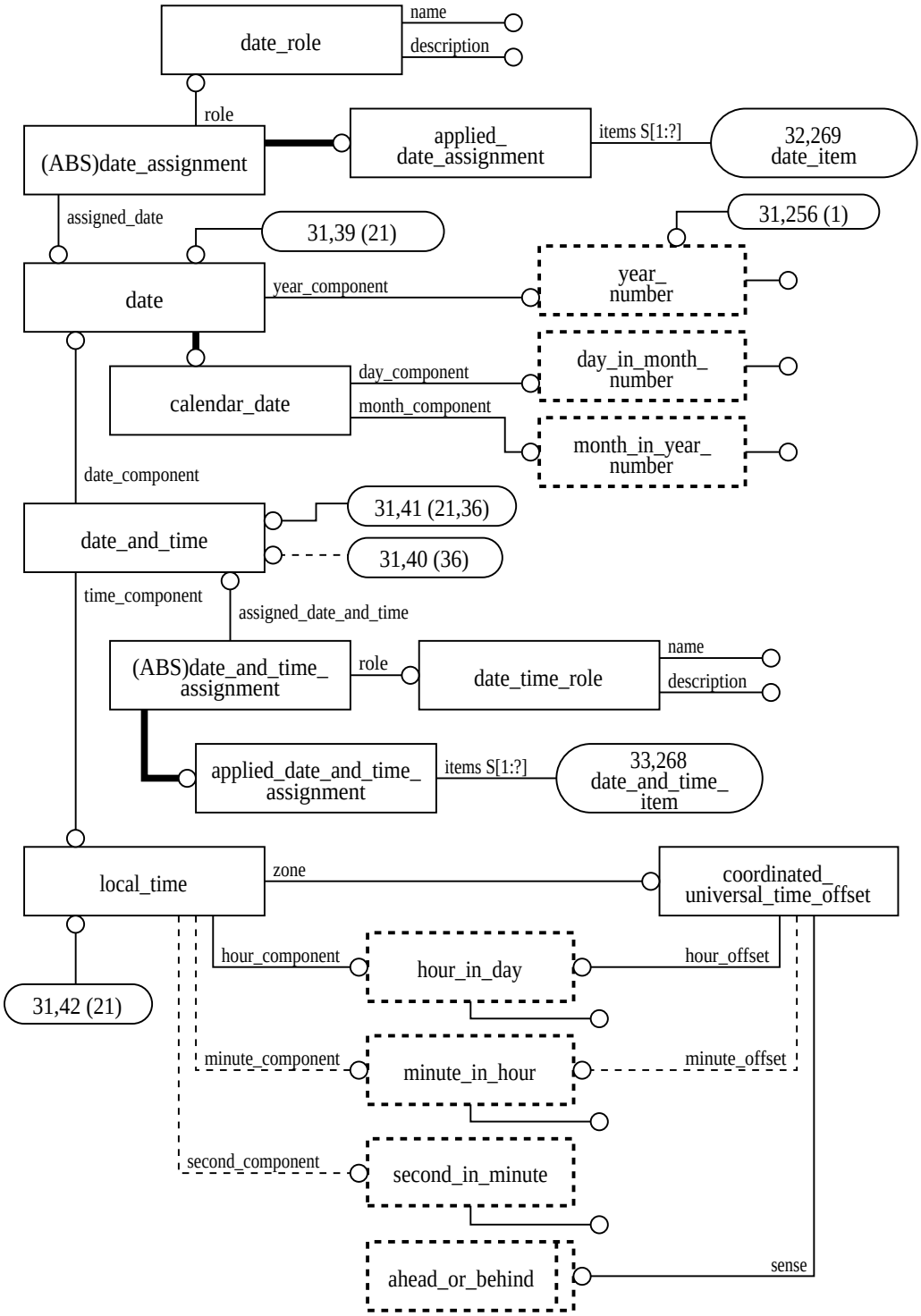


Figure H.31 – AIM diagram in EXPRESS_G: 31 of 124

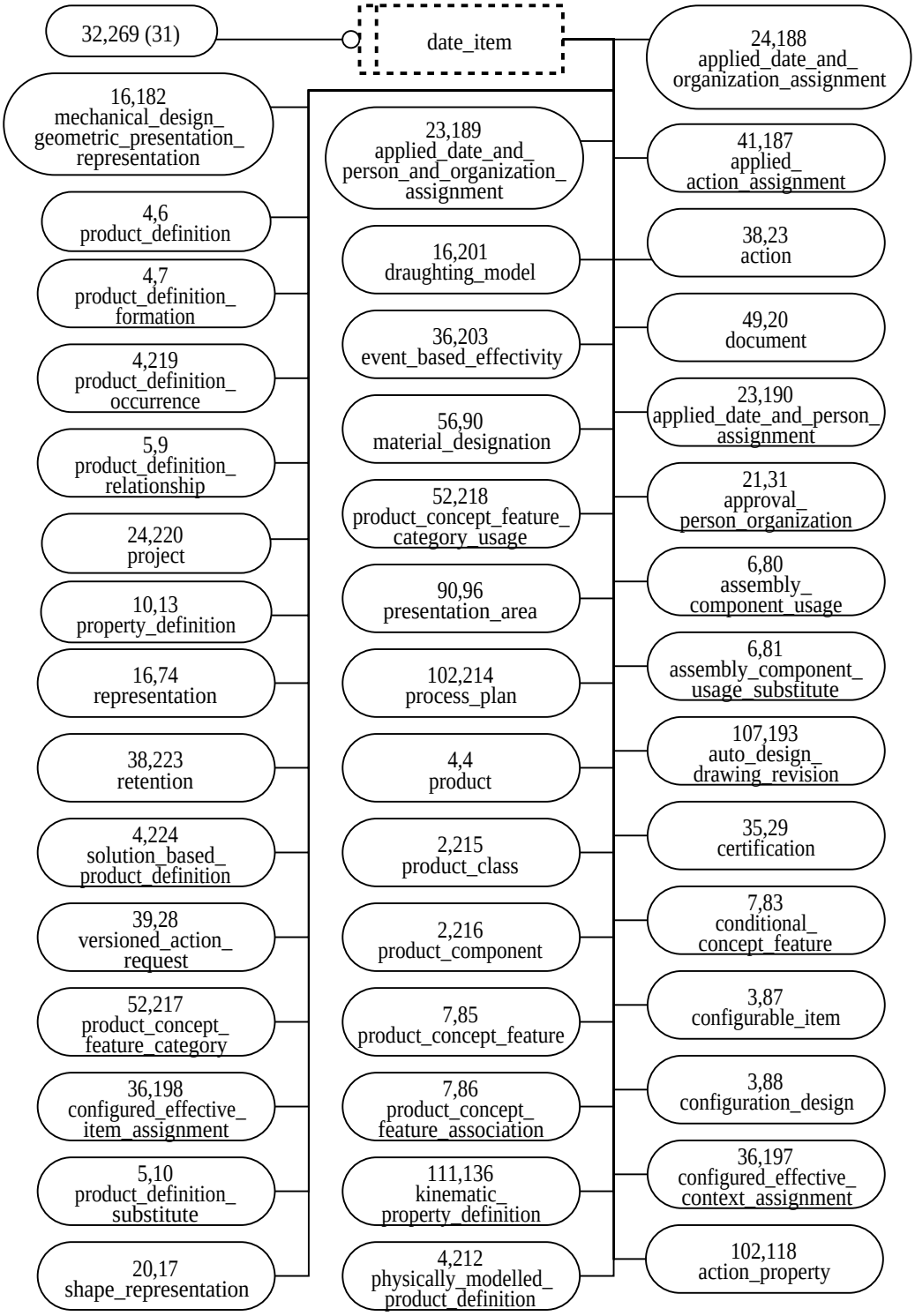


Figure H.32 – AIM diagram in EXPRESS_G: 32 of 124

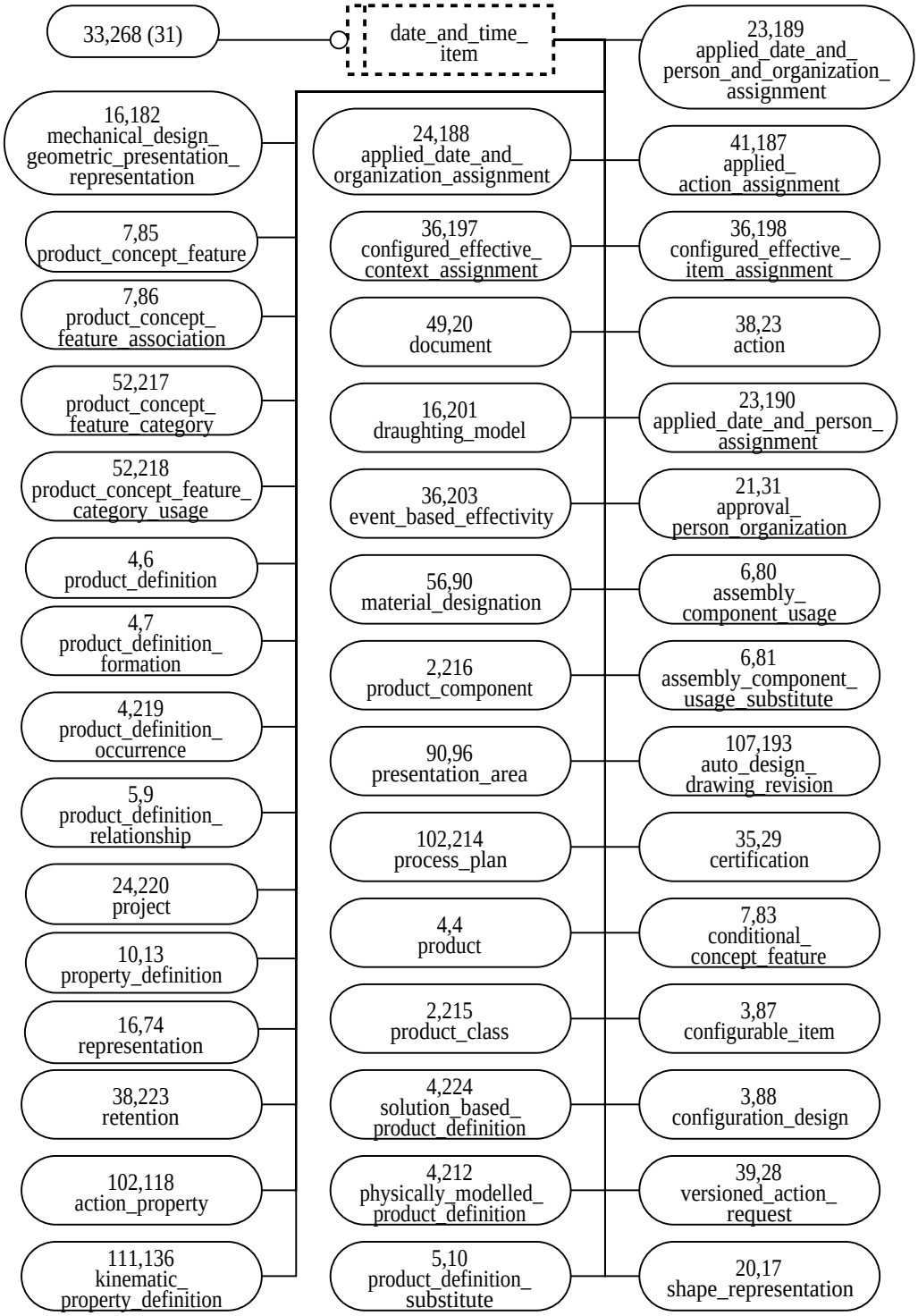


Figure H.33 – AIM diagram in EXPRESS_G: 33 of 124

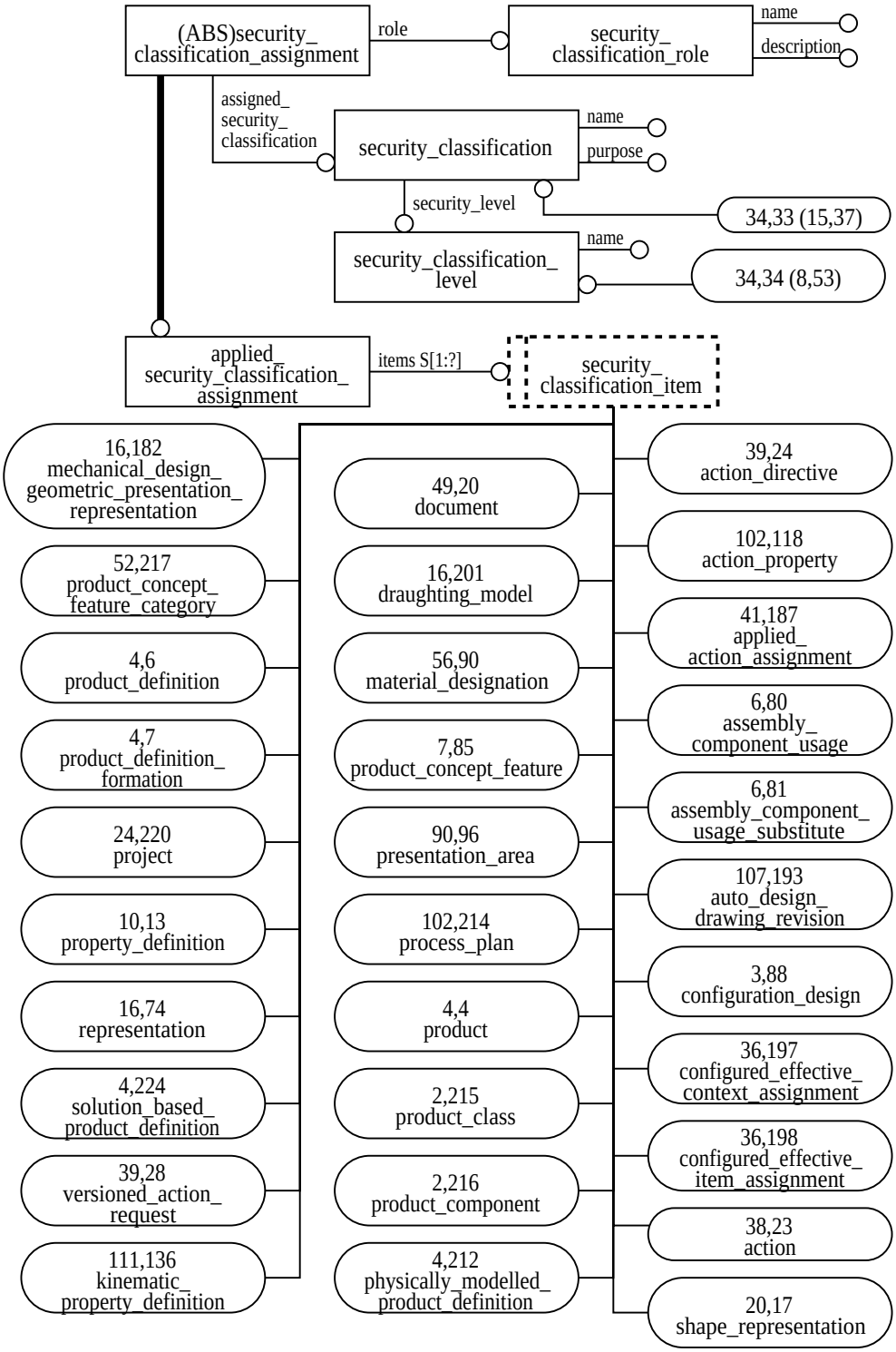


Figure H.34 – AIM diagram in EXPRESS_G: 34 of 124

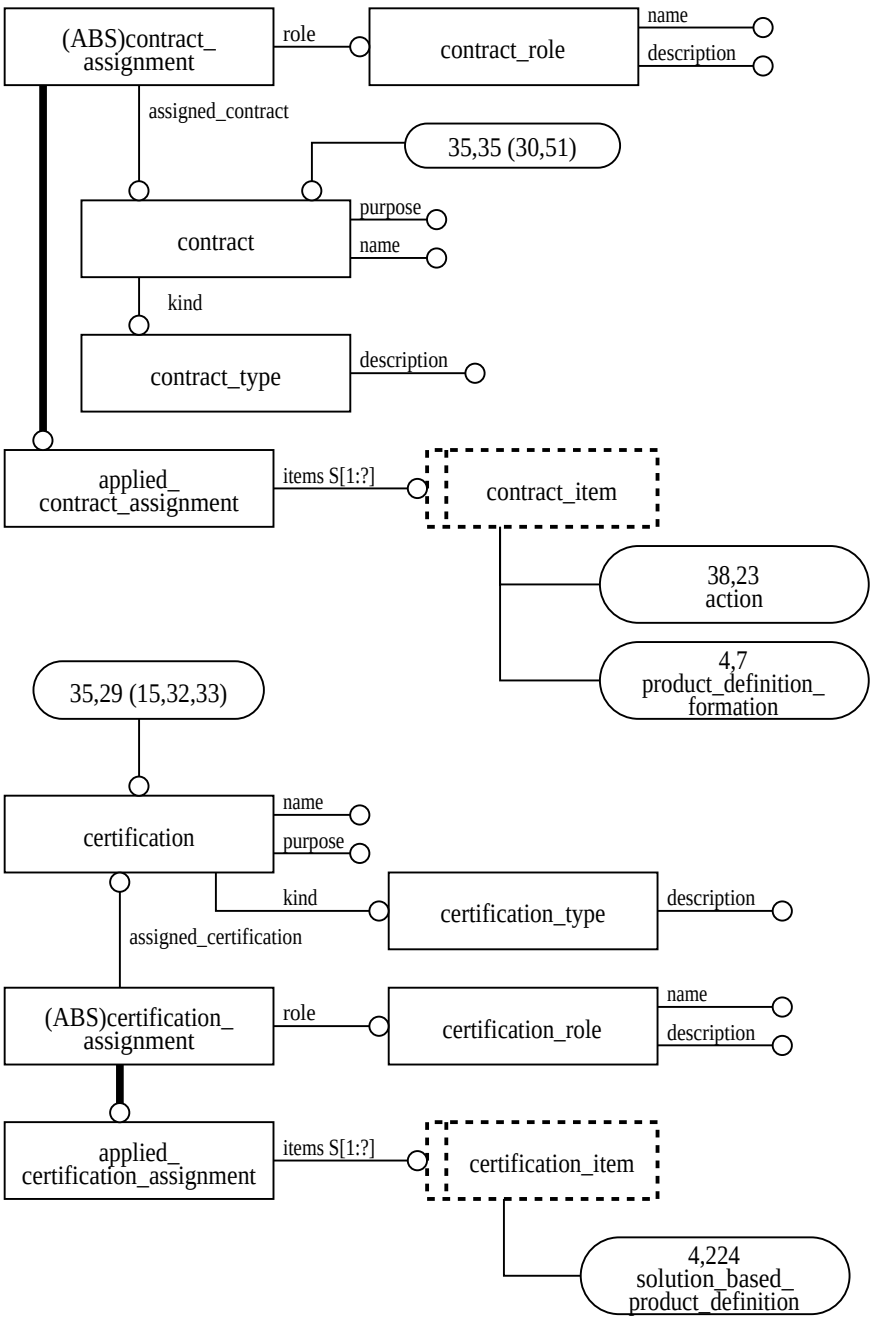


Figure H.35 – AIM diagram in EXPRESS_G: 35 of 124

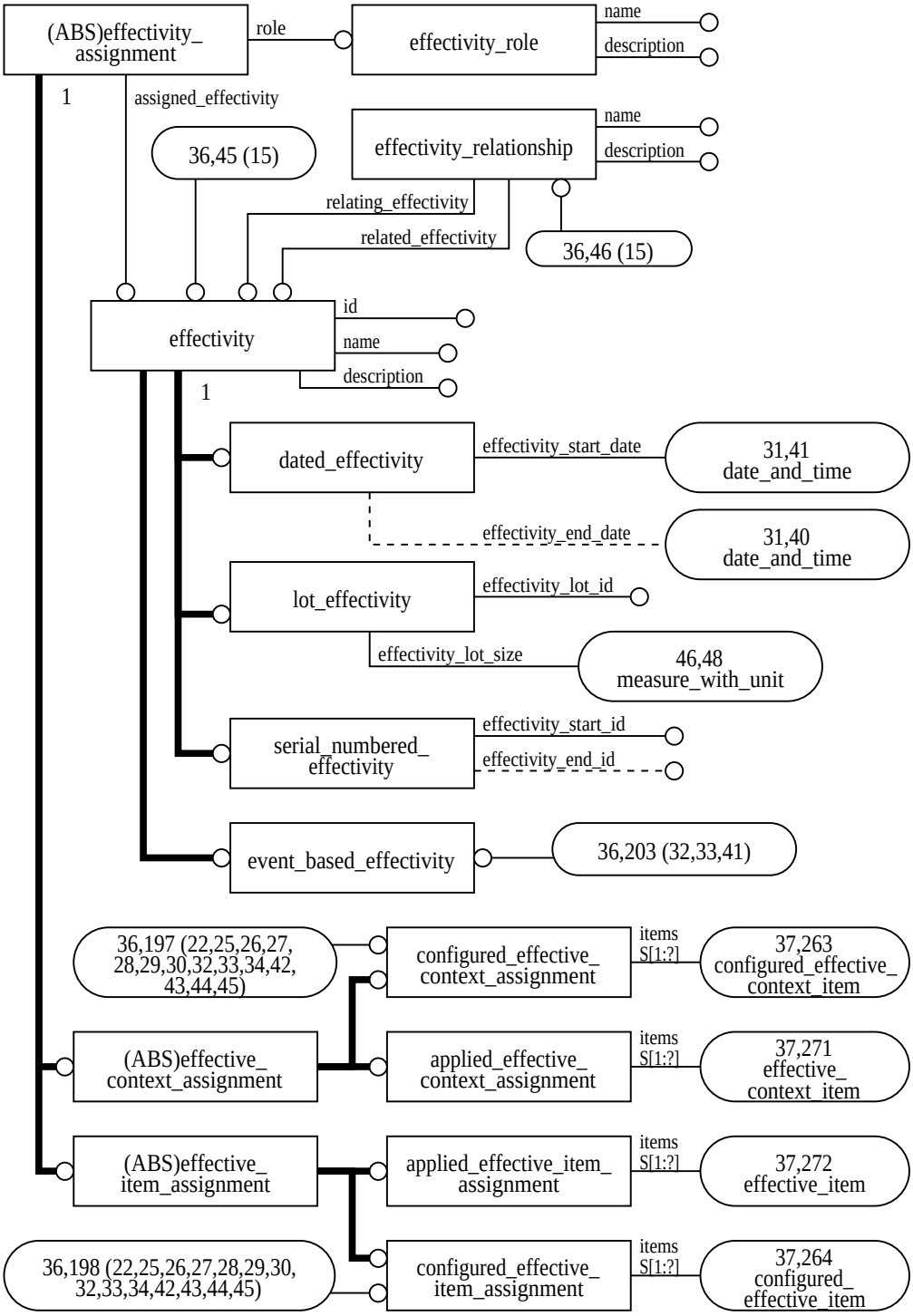


Figure H.36 – AIM diagram in EXPRESS_G: 36 of 124

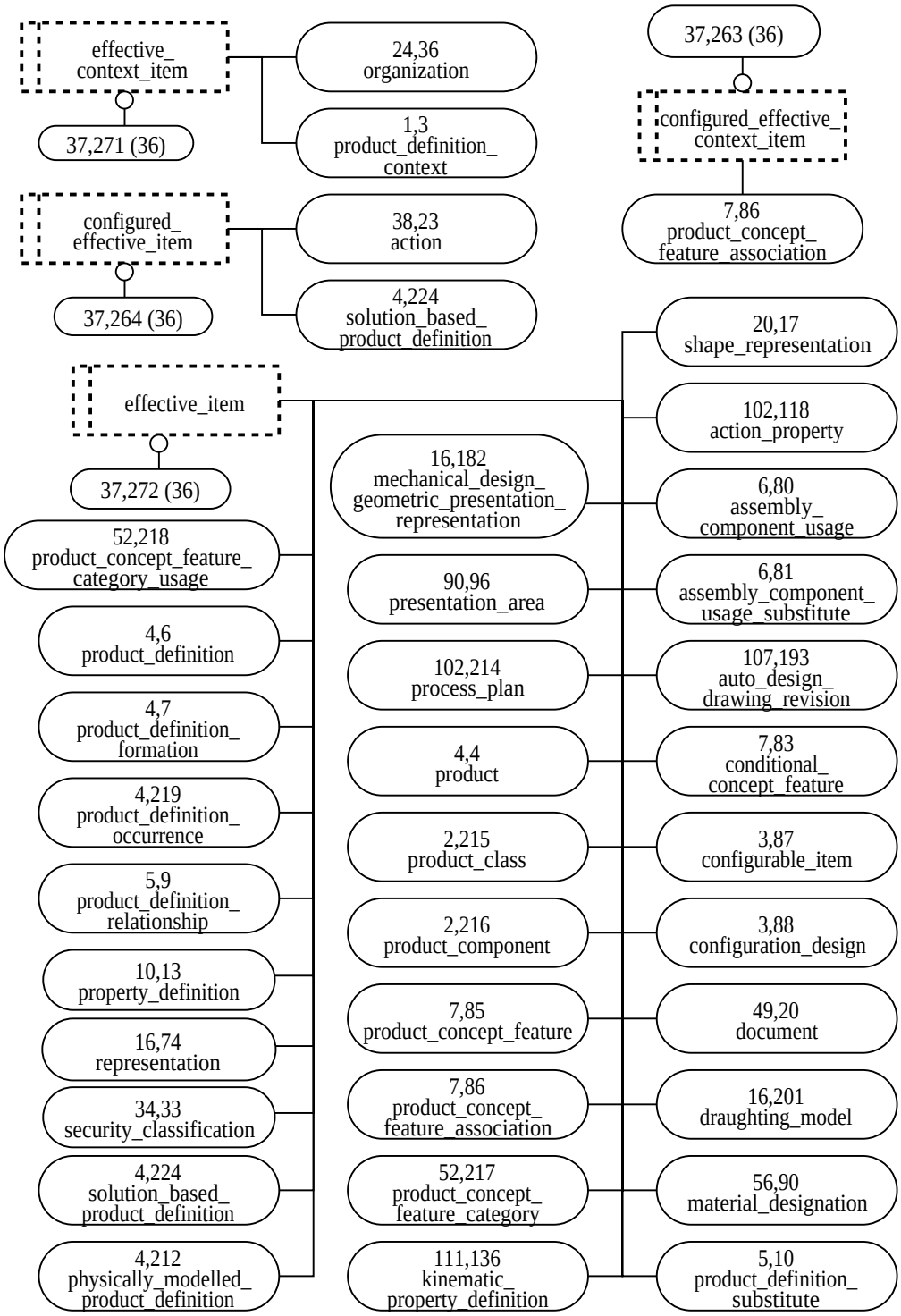


Figure H.37 – AIM diagram in EXPRESS_G: 37 of 124

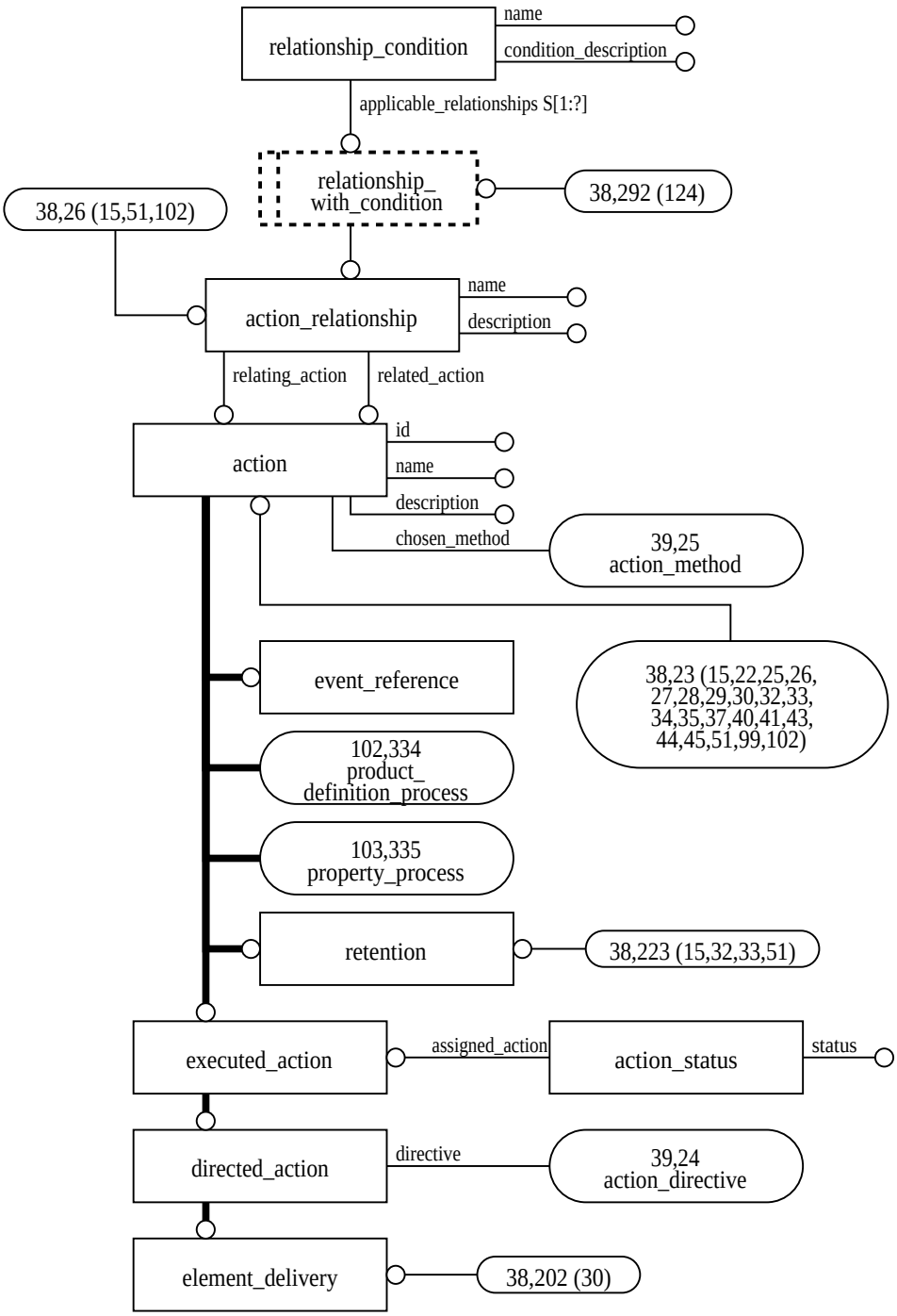


Figure H.38 – AIM diagram in EXPRESS_G: 38 of 124

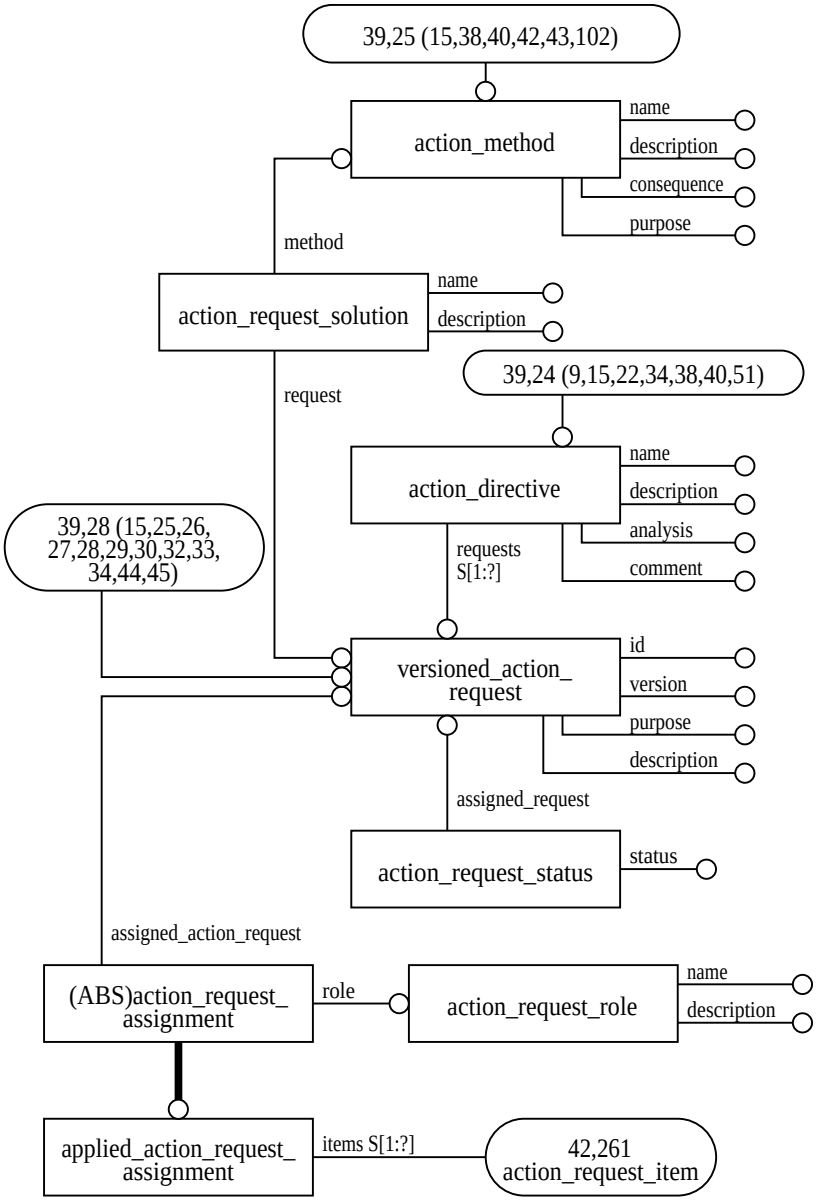


Figure H.39 – AIM diagram in EXPRESS_G: 39 of 124

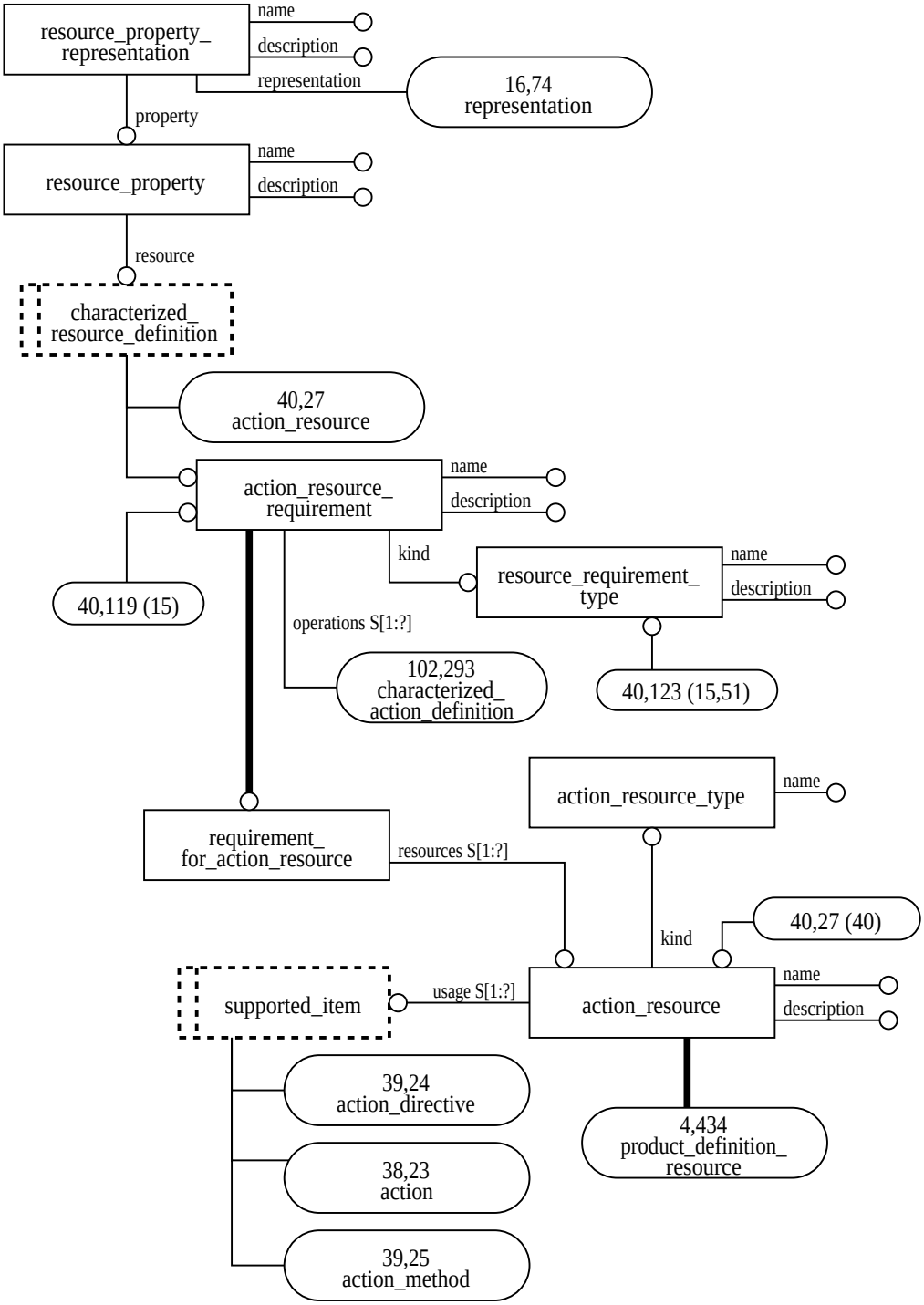


Figure H.40 – AIM diagram in EXPRESS_G: 40 of 124

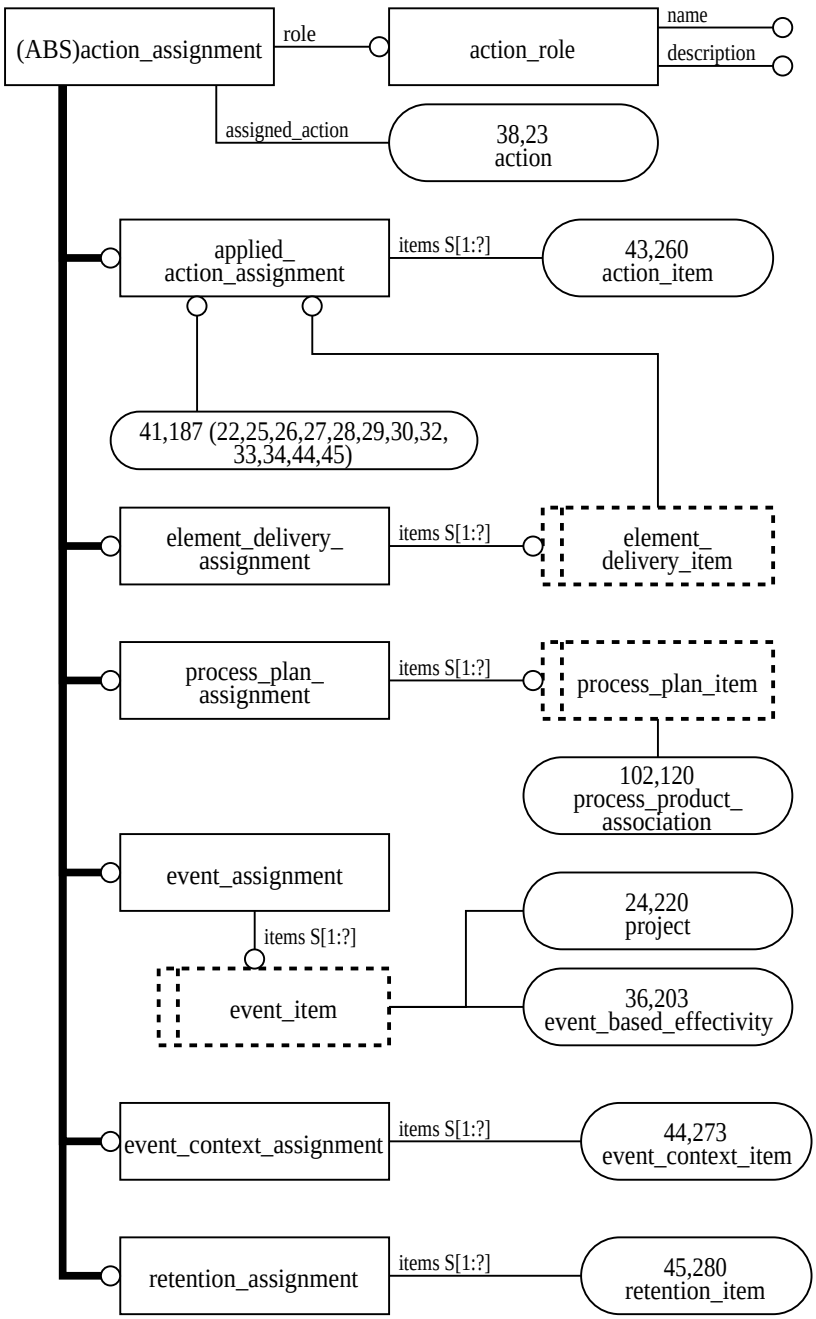


Figure H.41 – AIM diagram in EXPRESS_G: 41 of 124

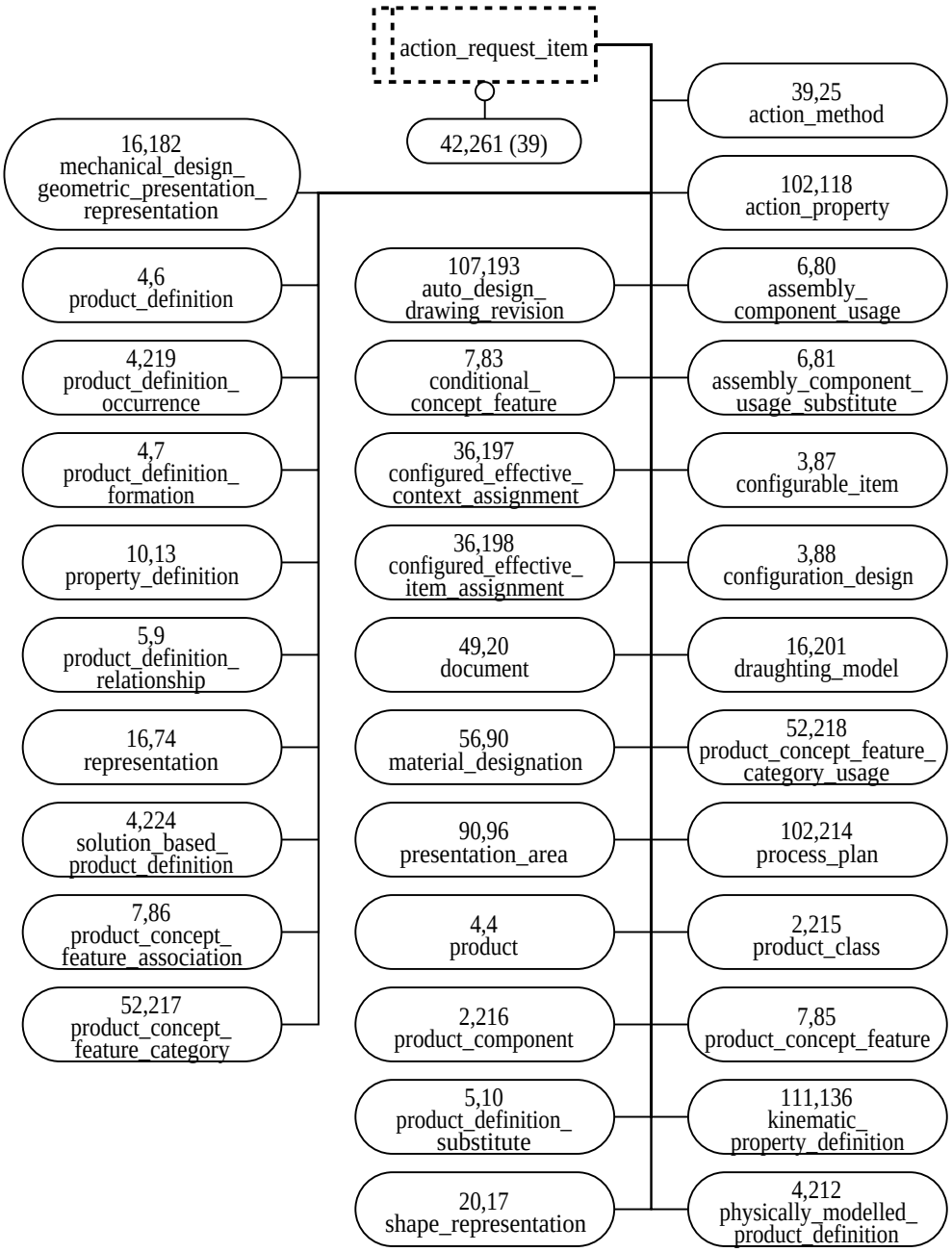


Figure H.42 – AIM diagram in EXPRESS_G: 42 of 124

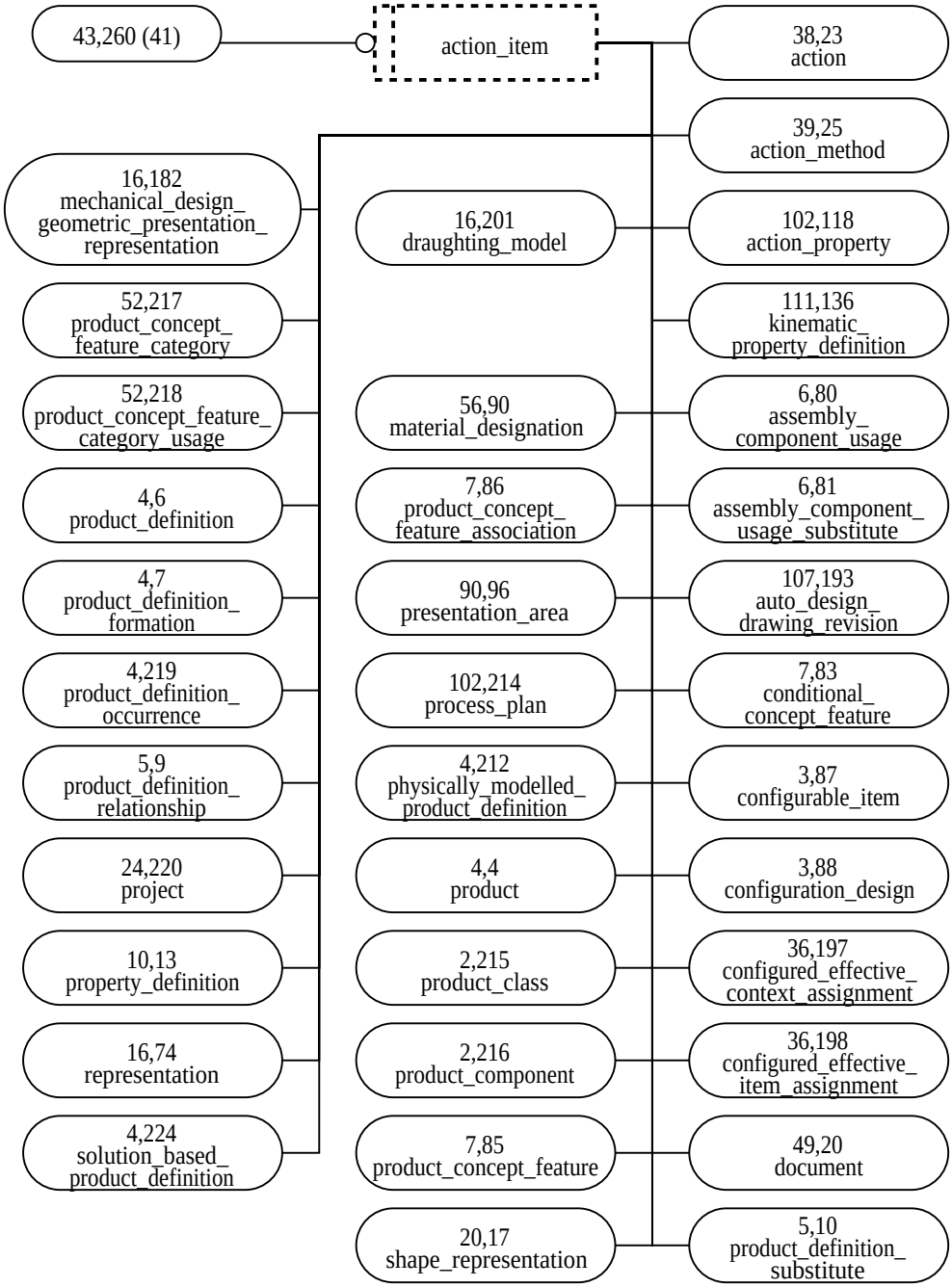


Figure H.43 – AIM diagram in EXPRESS_G: 43 of 124

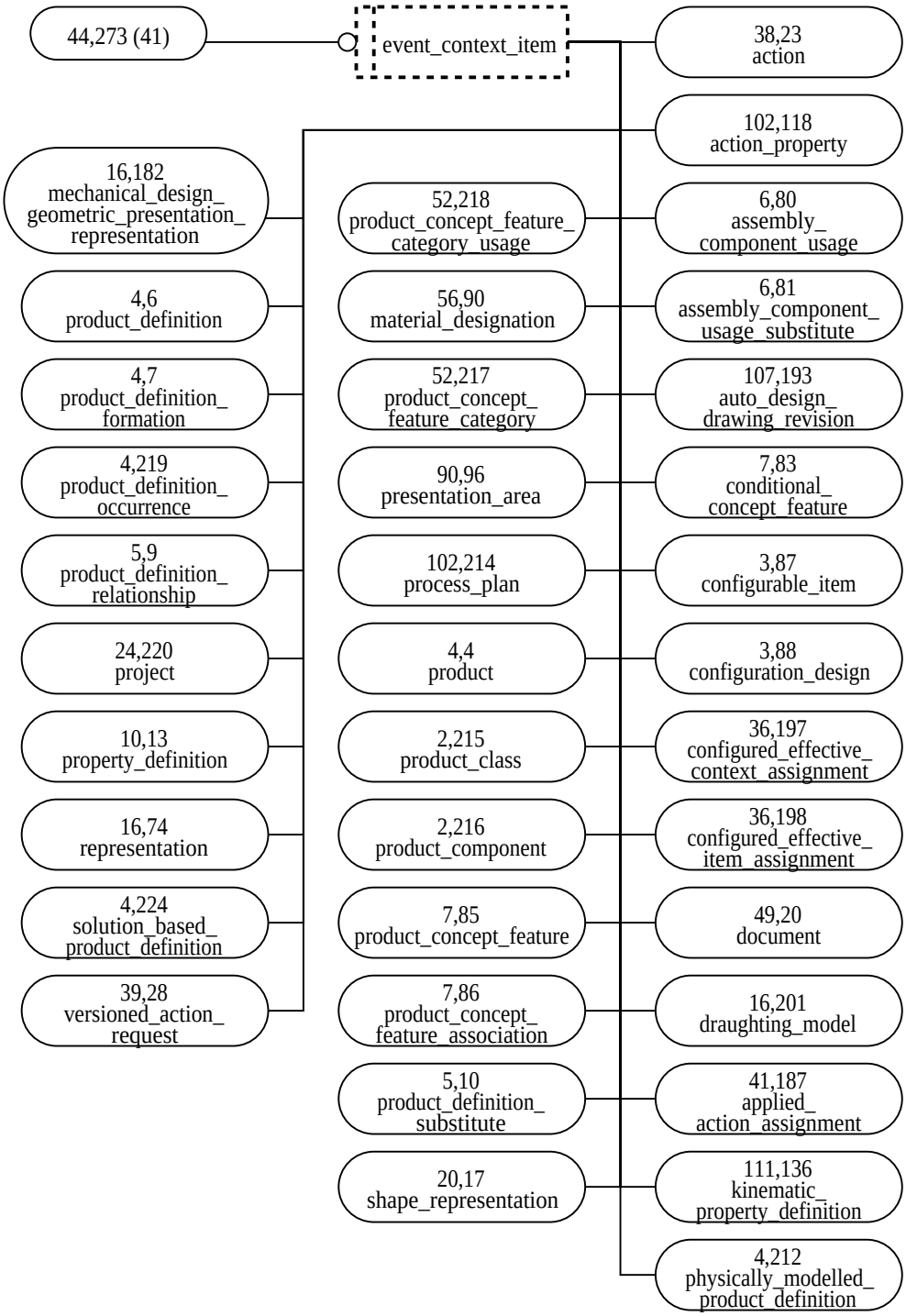


Figure H.44 – AIM diagram in EXPRESS_G: 44 of 124

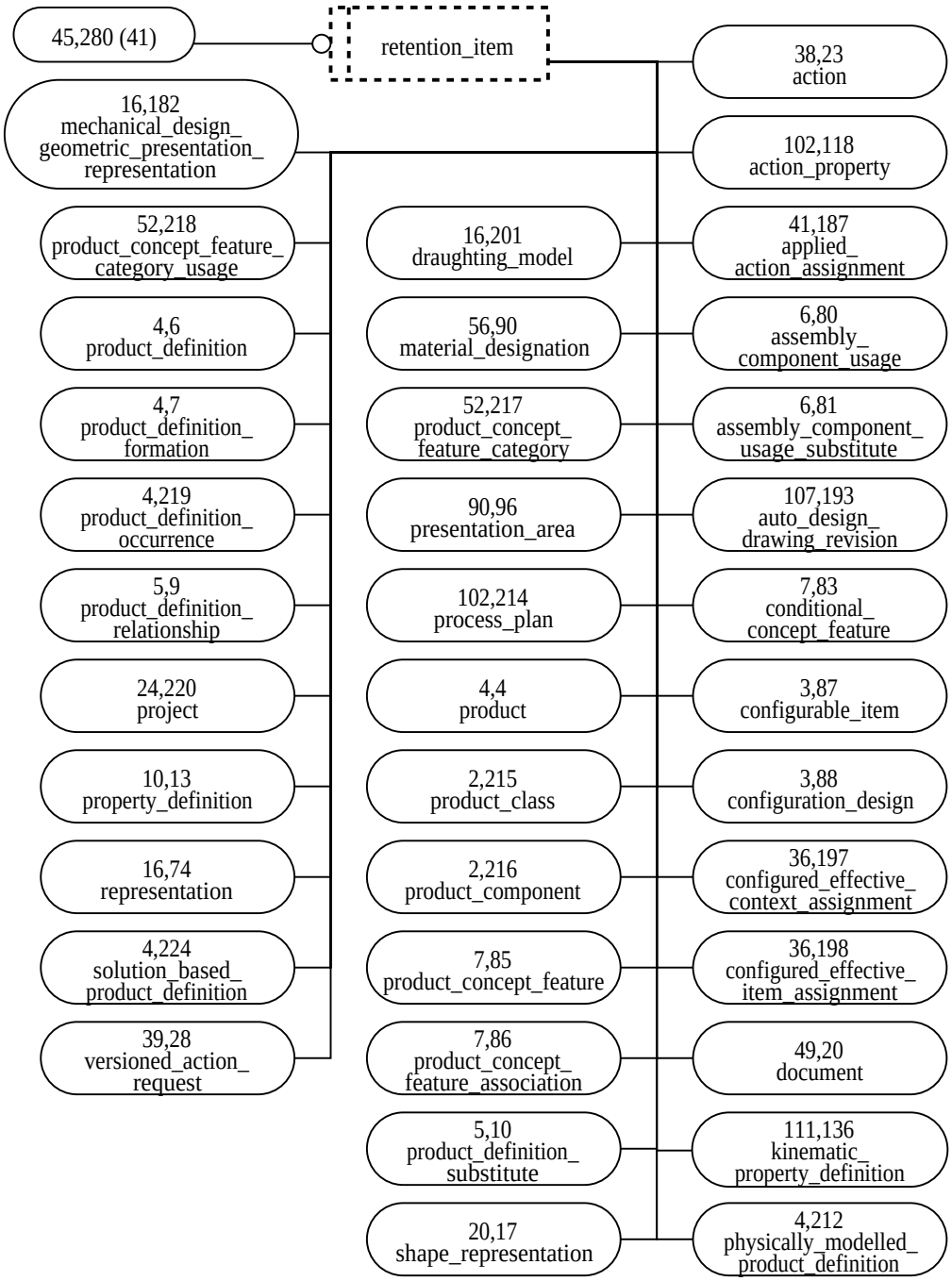


Figure H.45 – AIM diagram in EXPRESS_G: 45 of 124

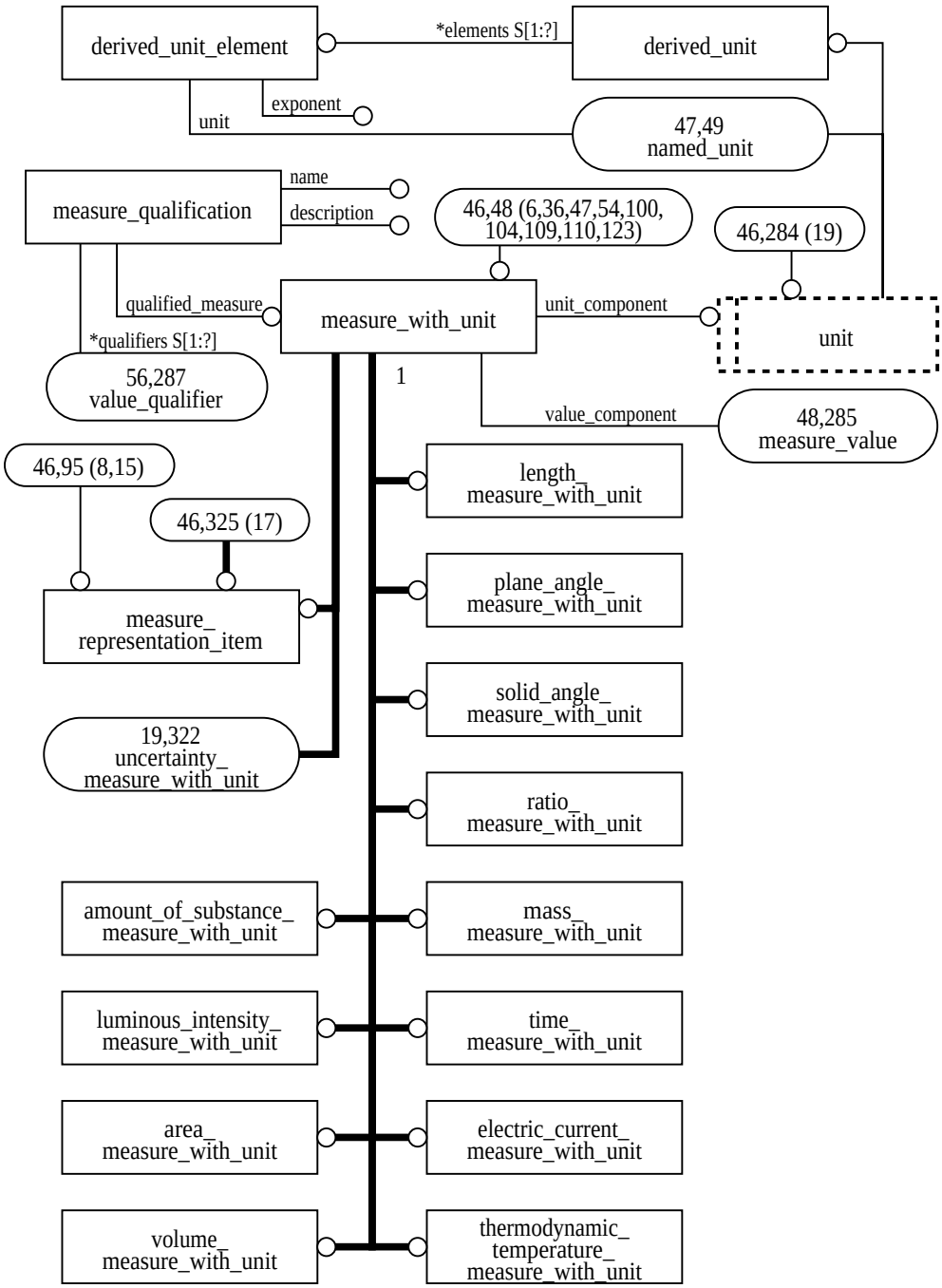


Figure H.46 – AIM diagram in EXPRESS_G: 46 of 124

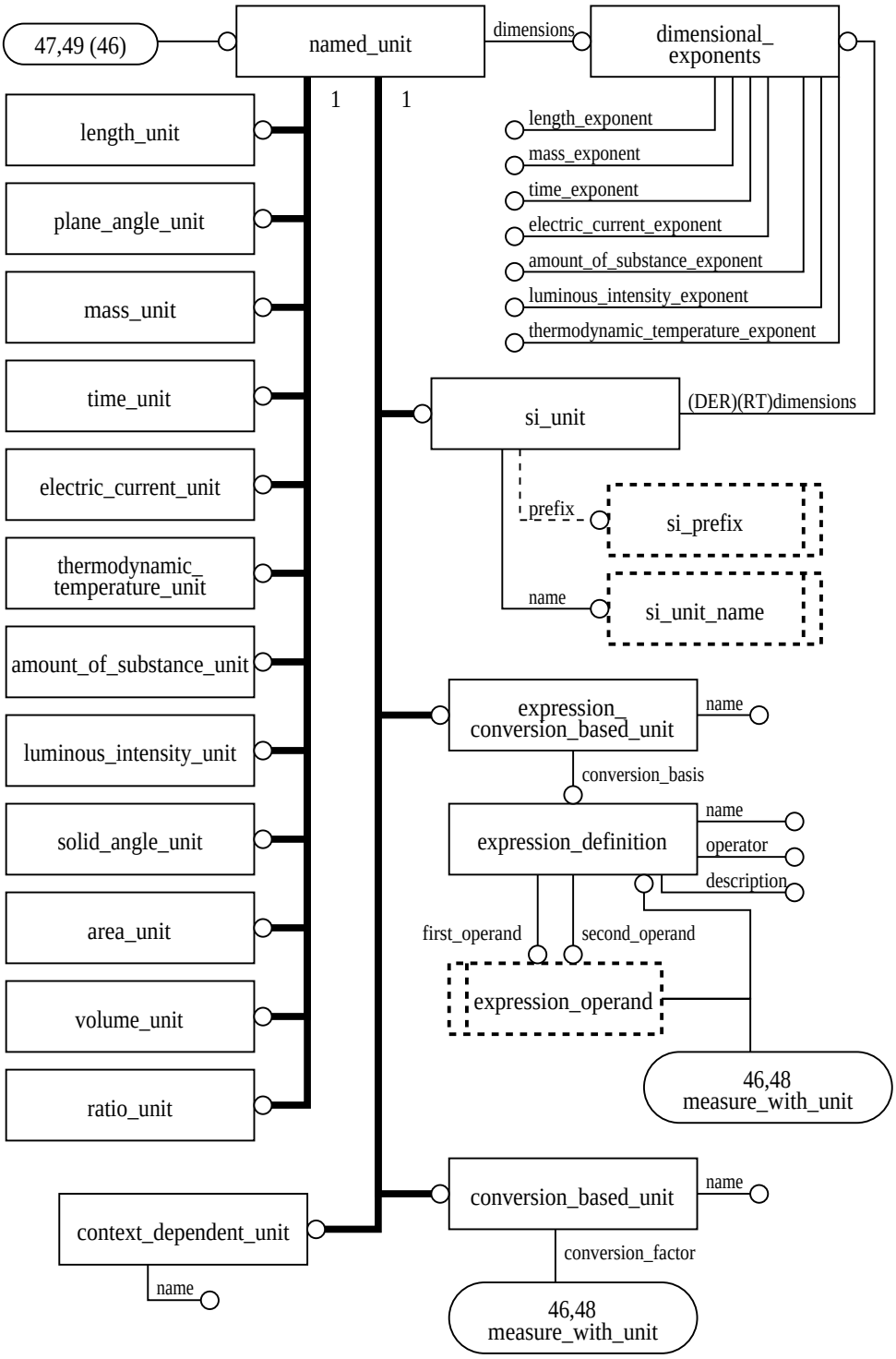


Figure H.47 – AIM diagram in EXPRESS_G: 47 of 124

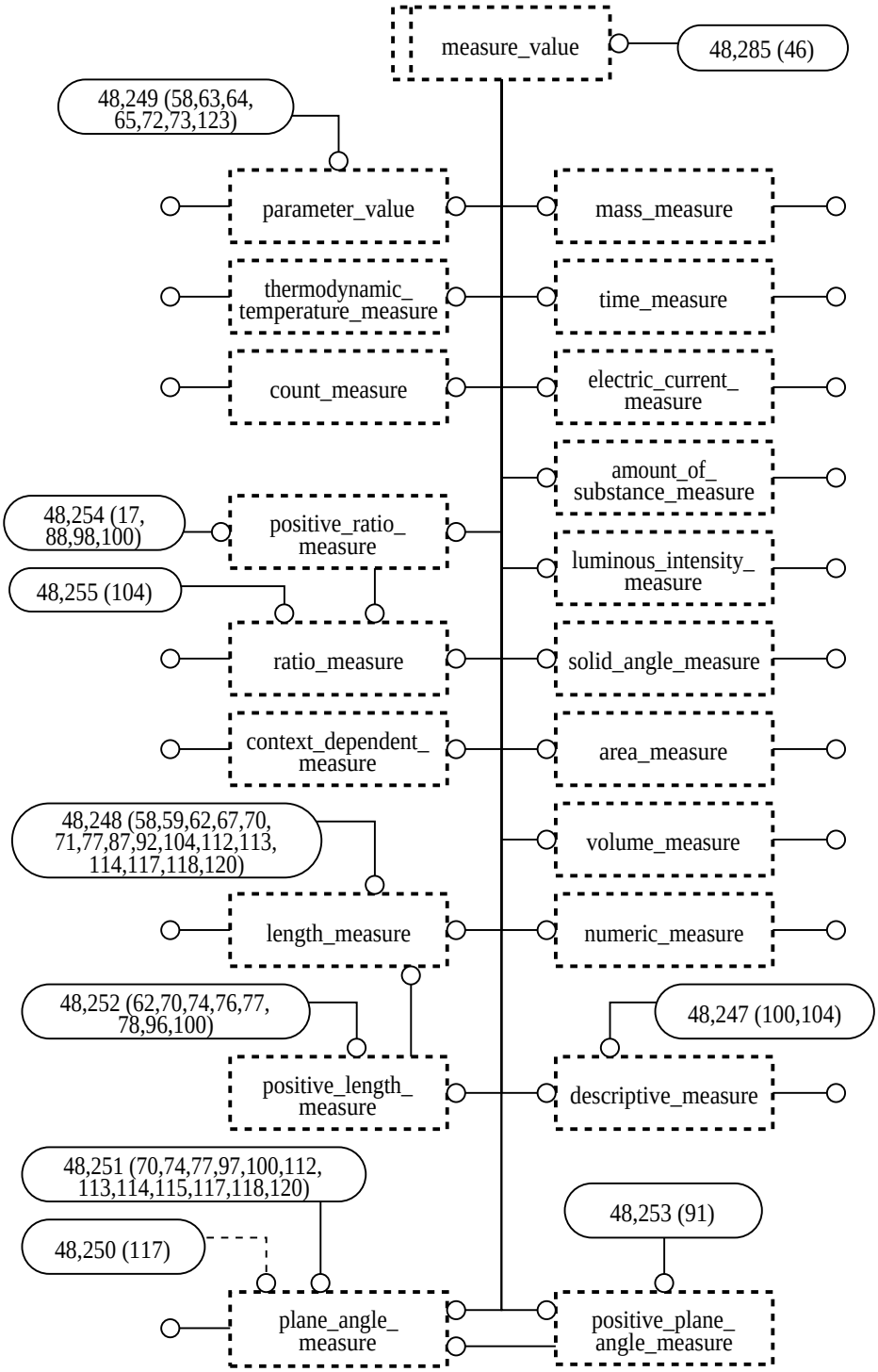


Figure H.48 – AIM diagram in EXPRESS_G: 48 of 124

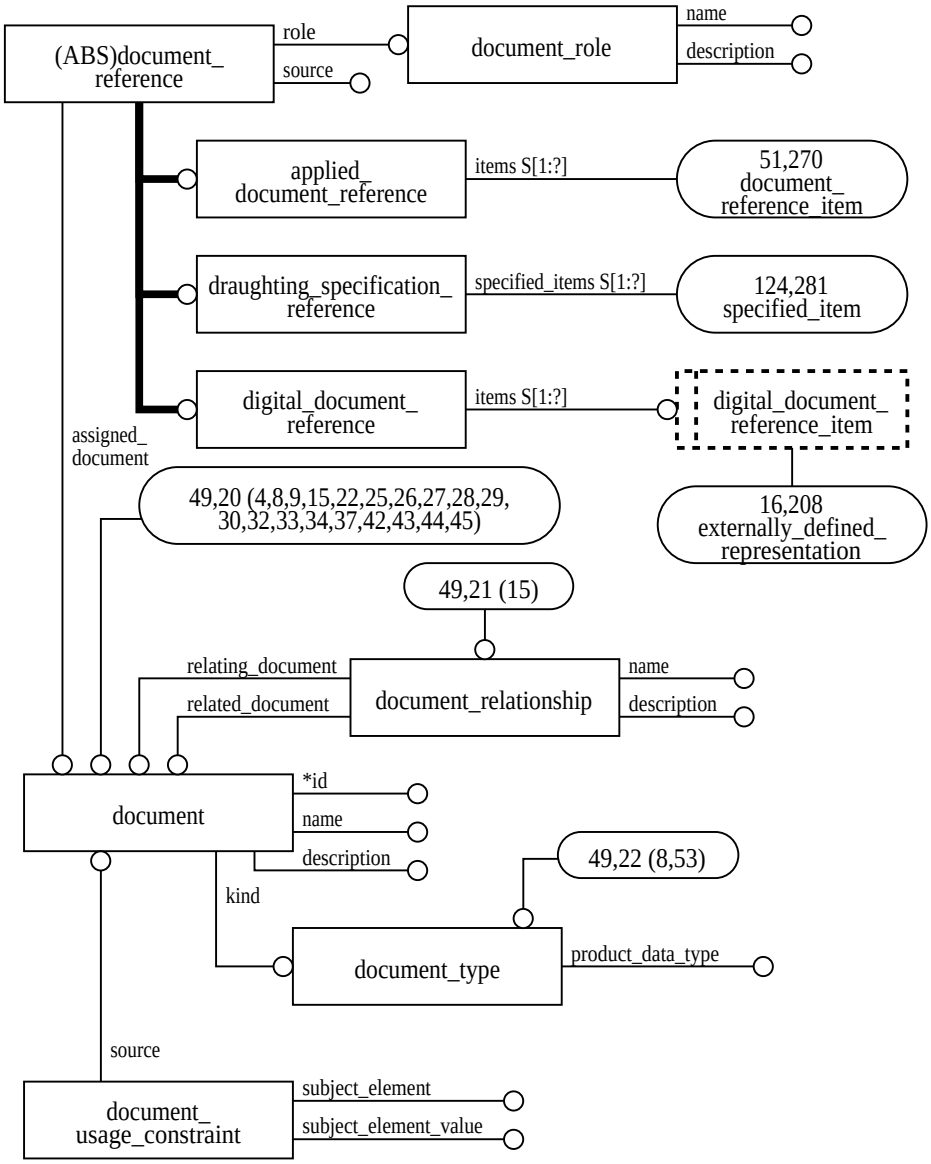


Figure H.49 – AIM diagram in EXPRESS_G: 49 of 124

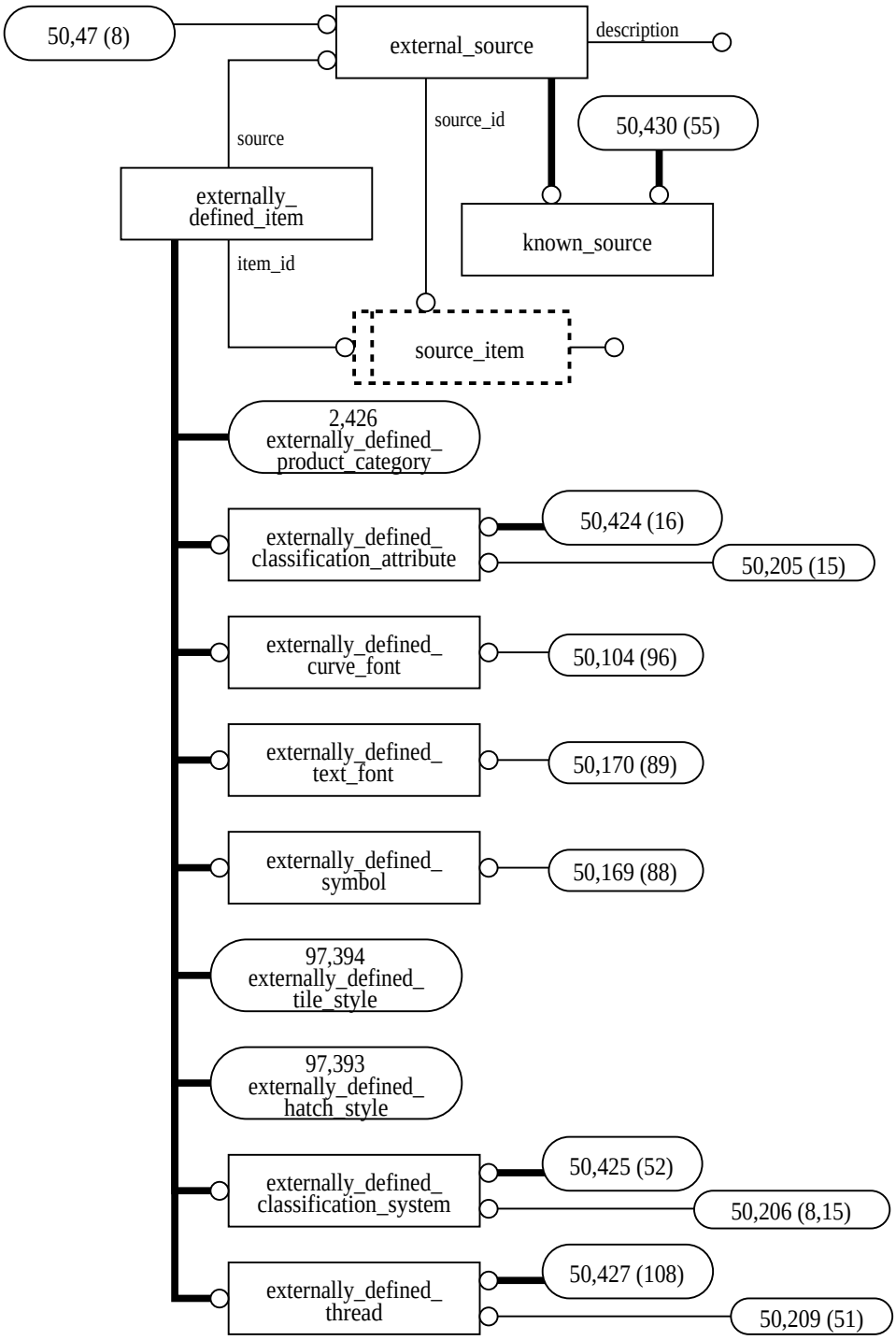


Figure H.50 – AIM diagram in EXPRESS_G: 50 of 124

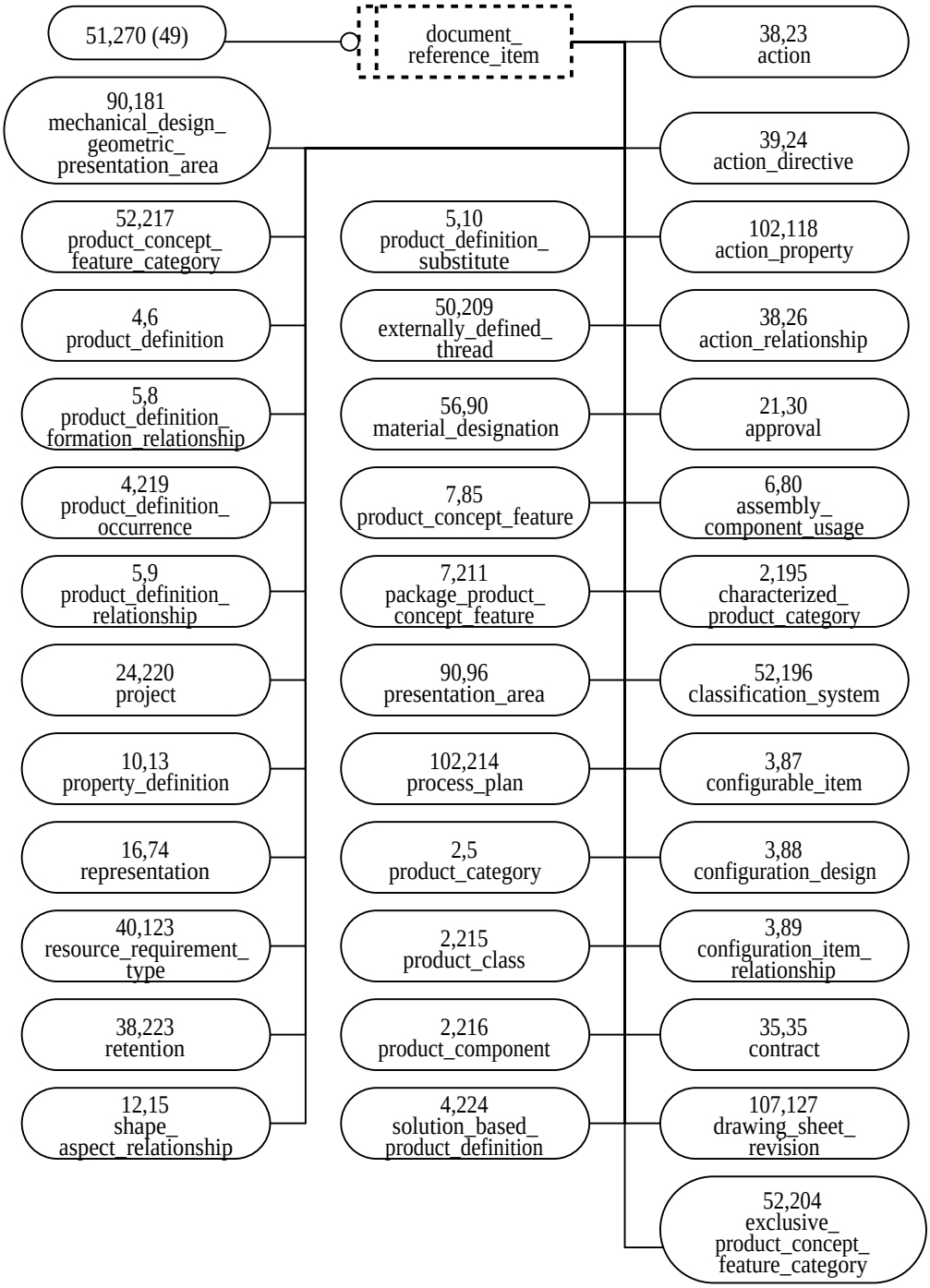


Figure H.51 – AIM diagram in EXPRESS_G: 51 of 124

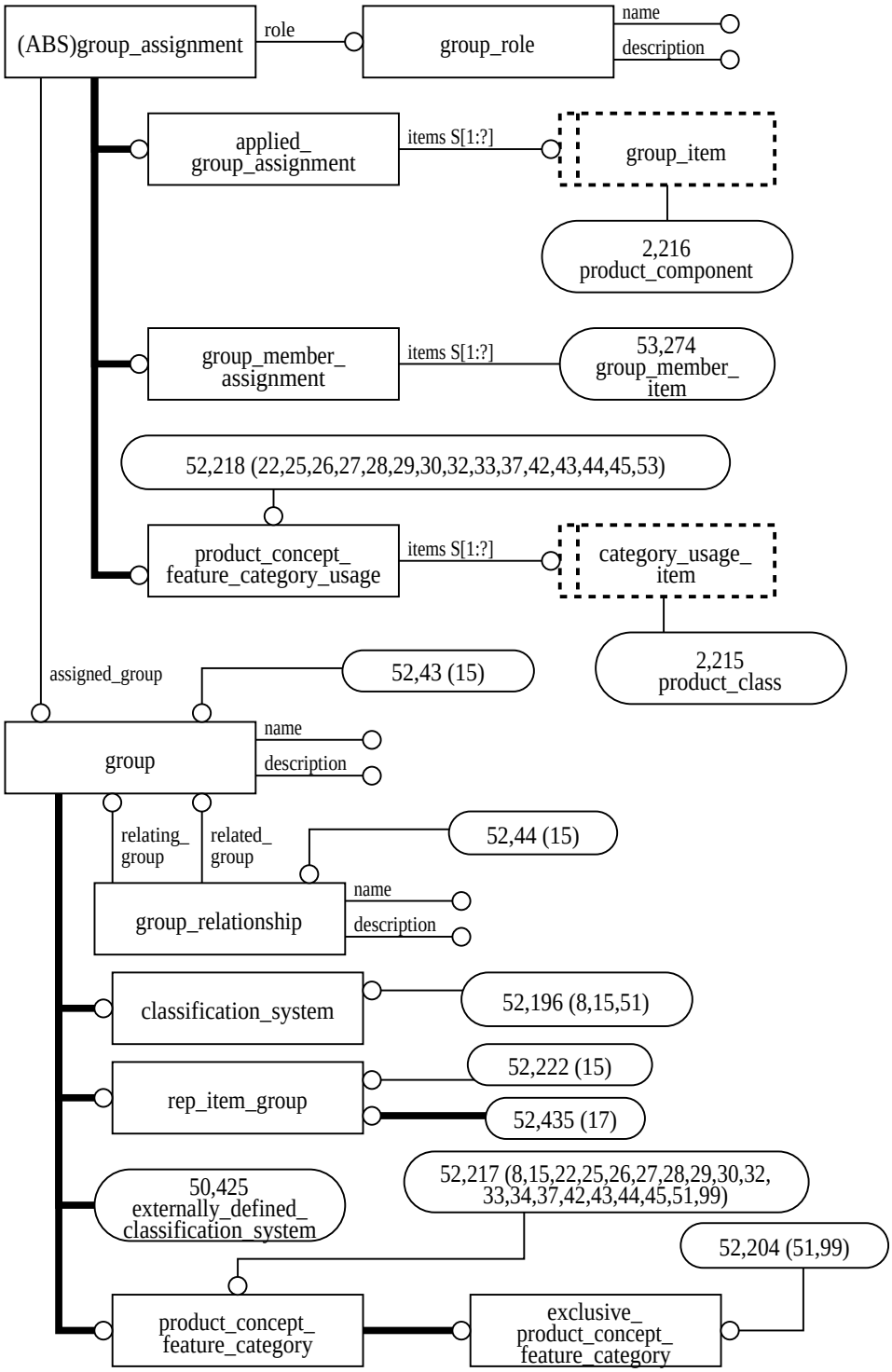


Figure H.52 – AIM diagram in EXPRESS_G: 52 of 124

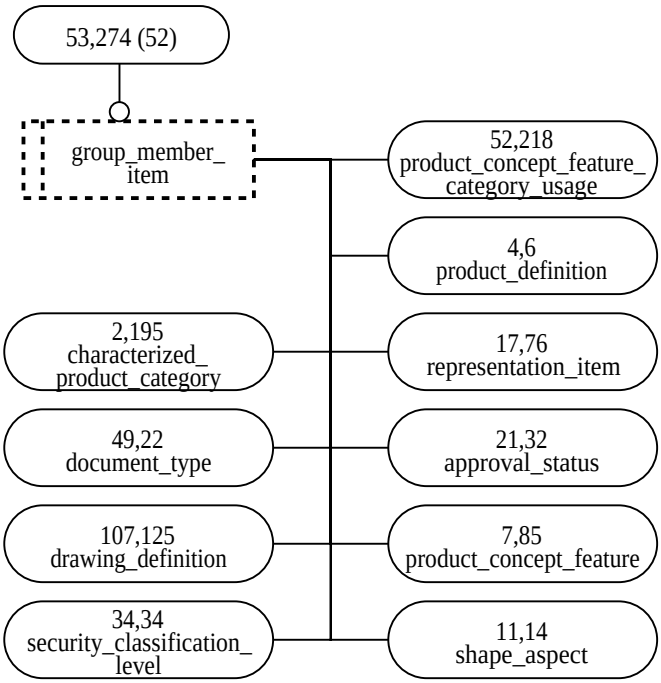


Figure H.53 – AIM diagram in EXPRESS_G: 53 of 124

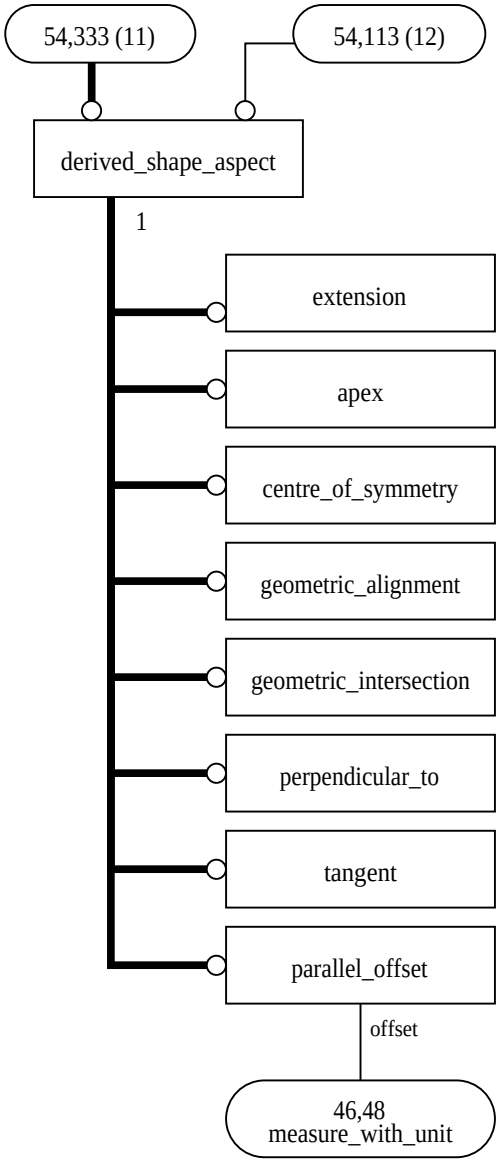


Figure H.54 – AIM diagram in EXPRESS_G: 54 of 124

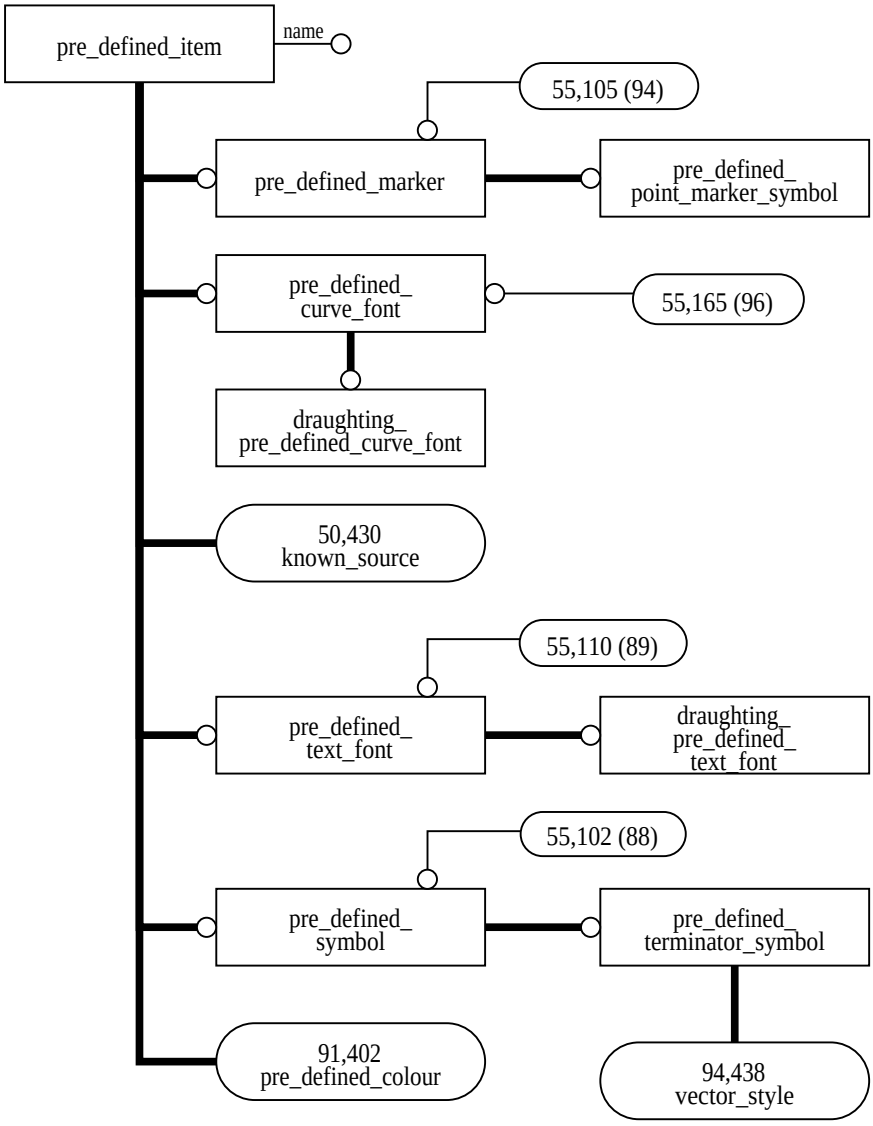


Figure H.55 – AIM diagram in EXPRESS_G: 55 of 124

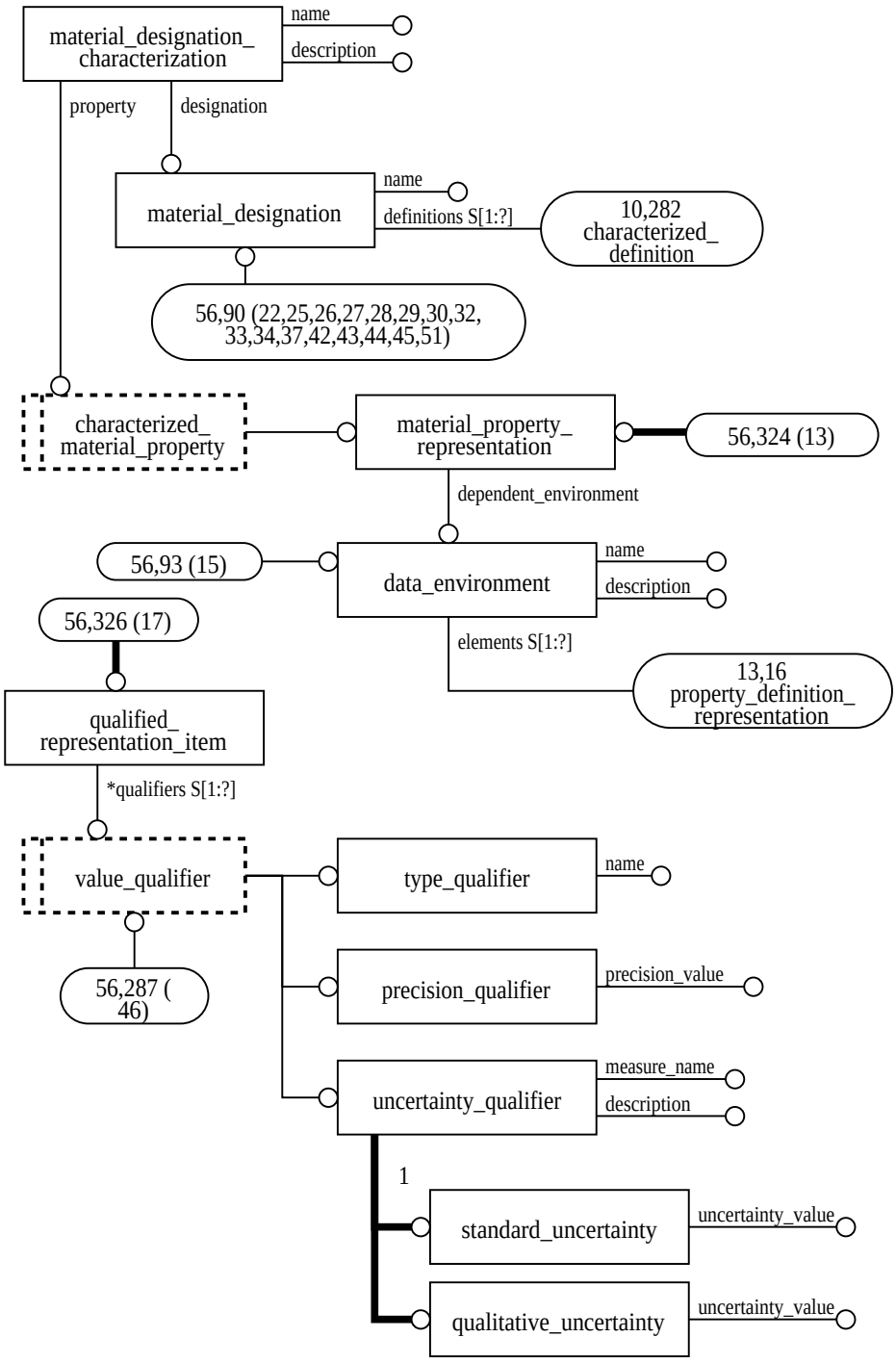


Figure H.56 – AIM diagram in EXPRESS_G: 56 of 124

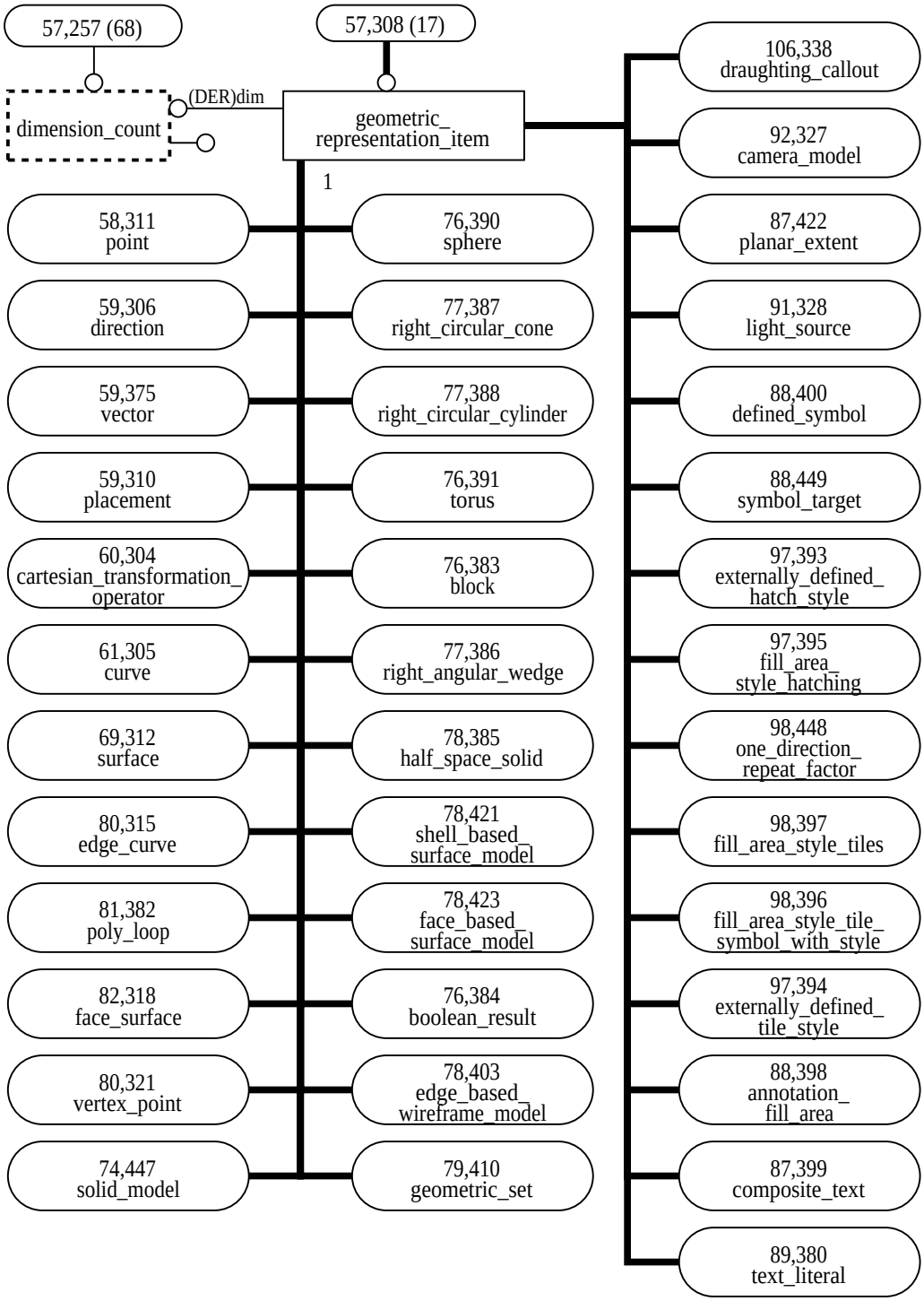


Figure H.57 – AIM diagram in EXPRESS_G: 57 of 124

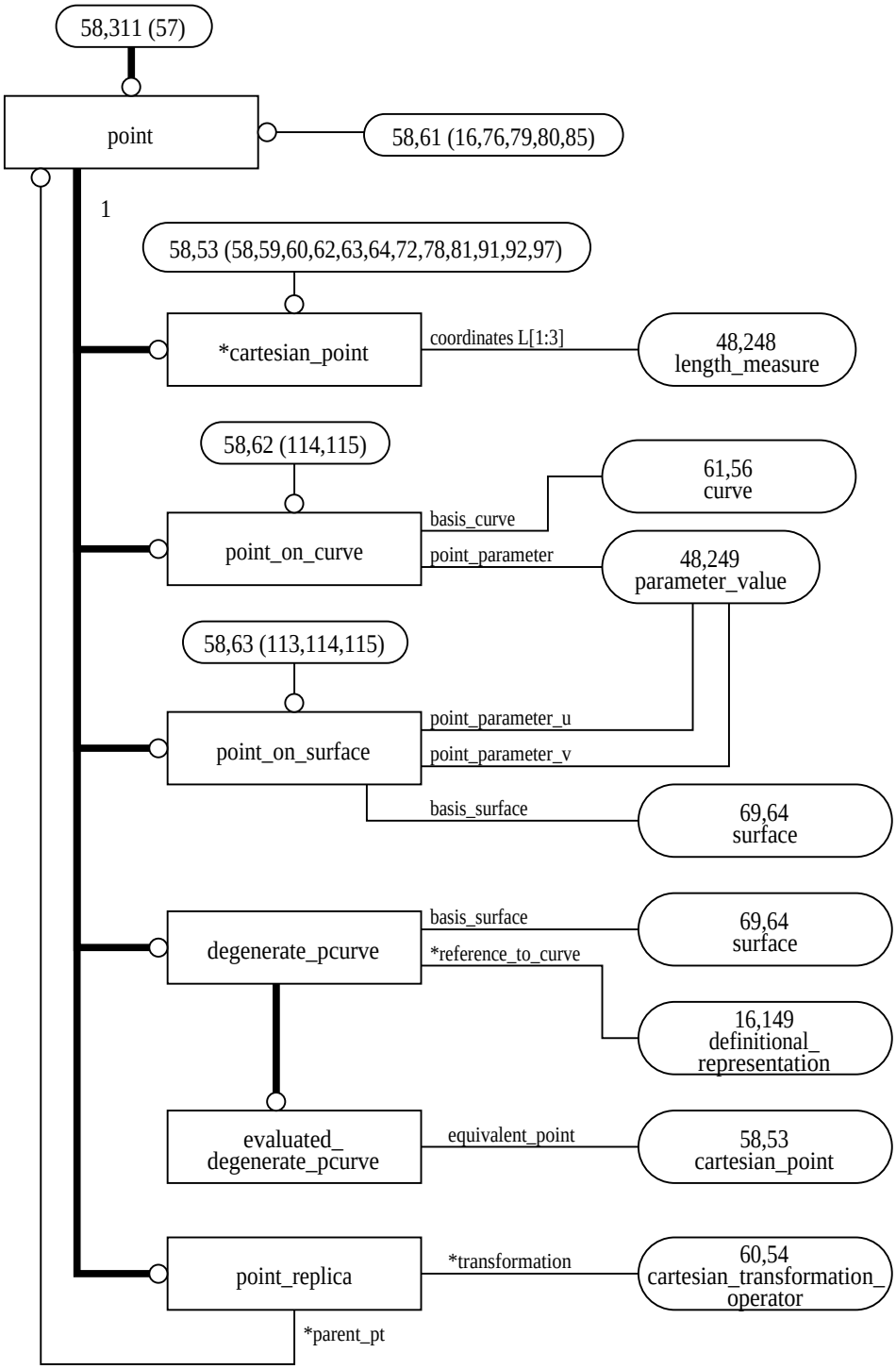


Figure H.58 – AIM diagram in EXPRESS_G: 58 of 124

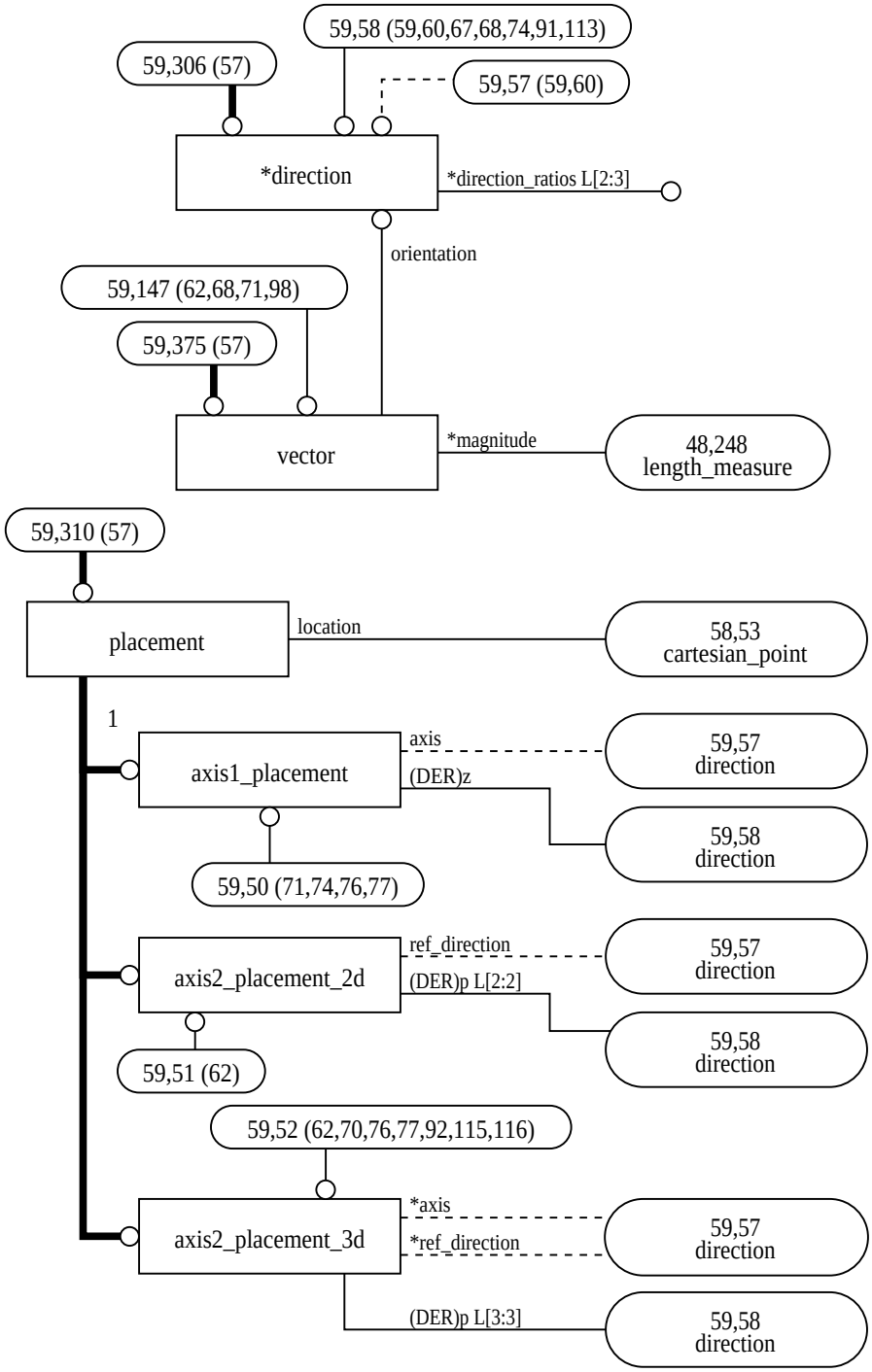


Figure H.59 – AIM diagram in EXPRESS_G: 59 of 124

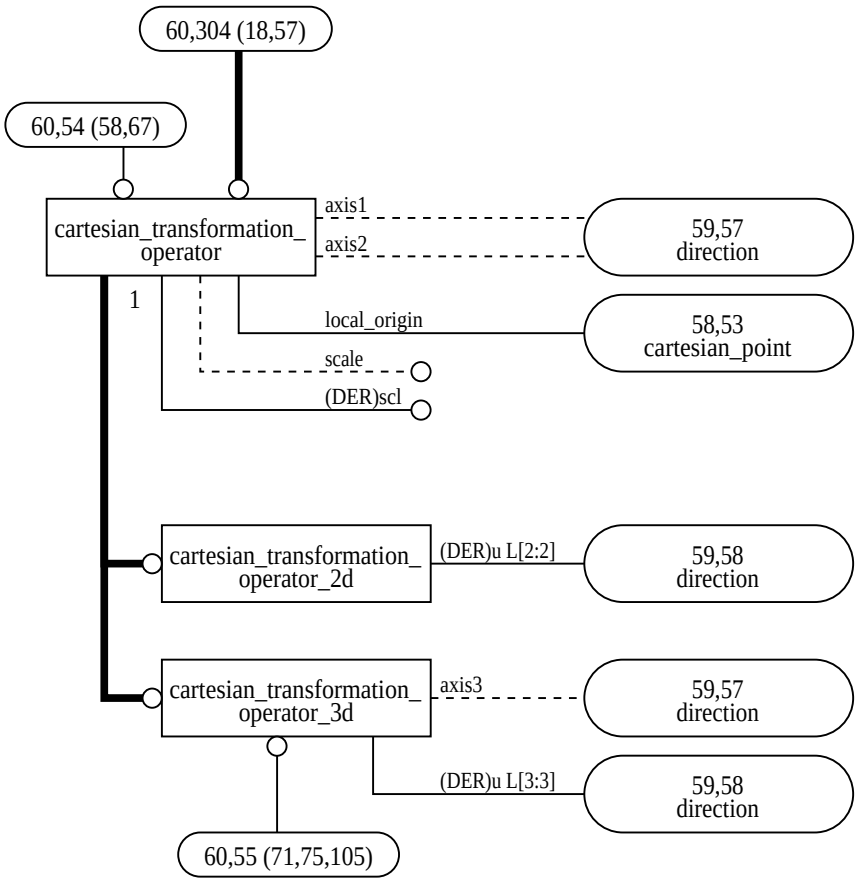


Figure H.60 – AIM diagram in EXPRESS_G: 60 of 124

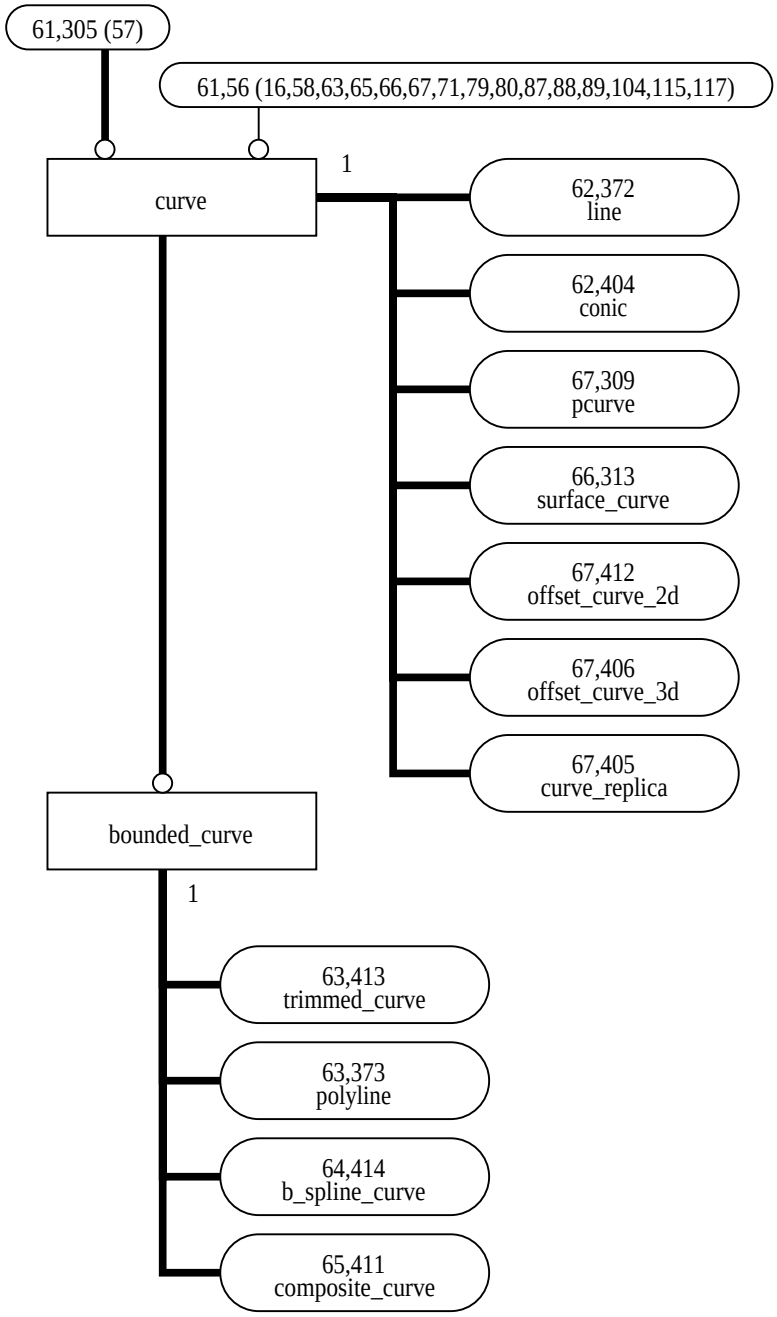


Figure H.61 – AIM diagram in EXPRESS_G: 61 of 124

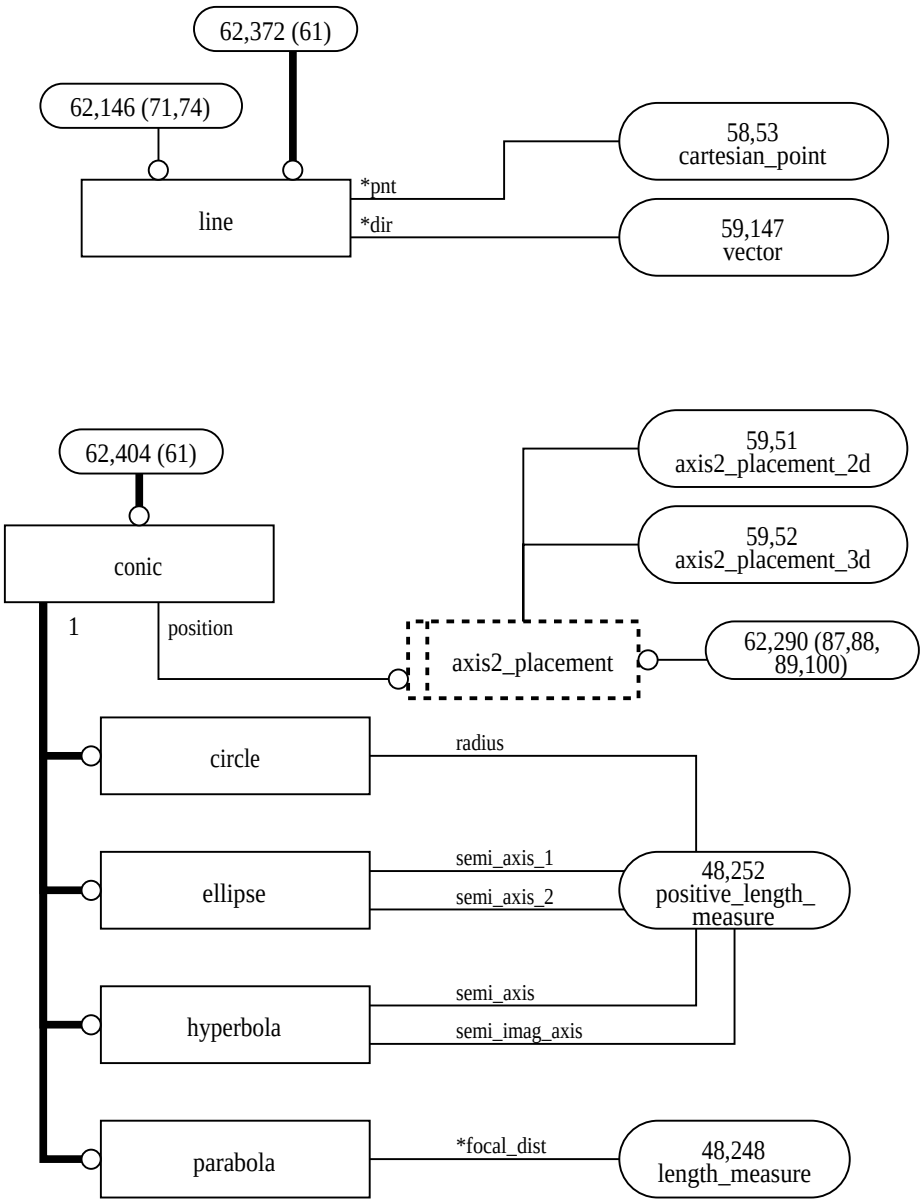


Figure H.62 – AIM diagram in EXPRESS_G: 62 of 124

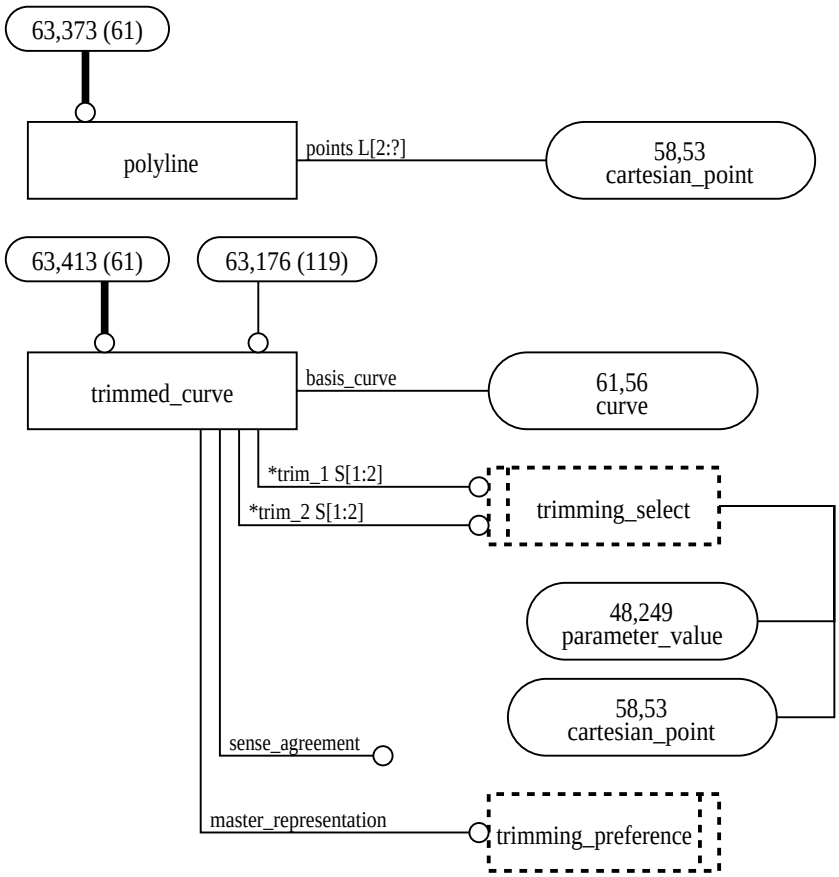


Figure H.63 – AIM diagram in EXPRESS_G: 63 of 124

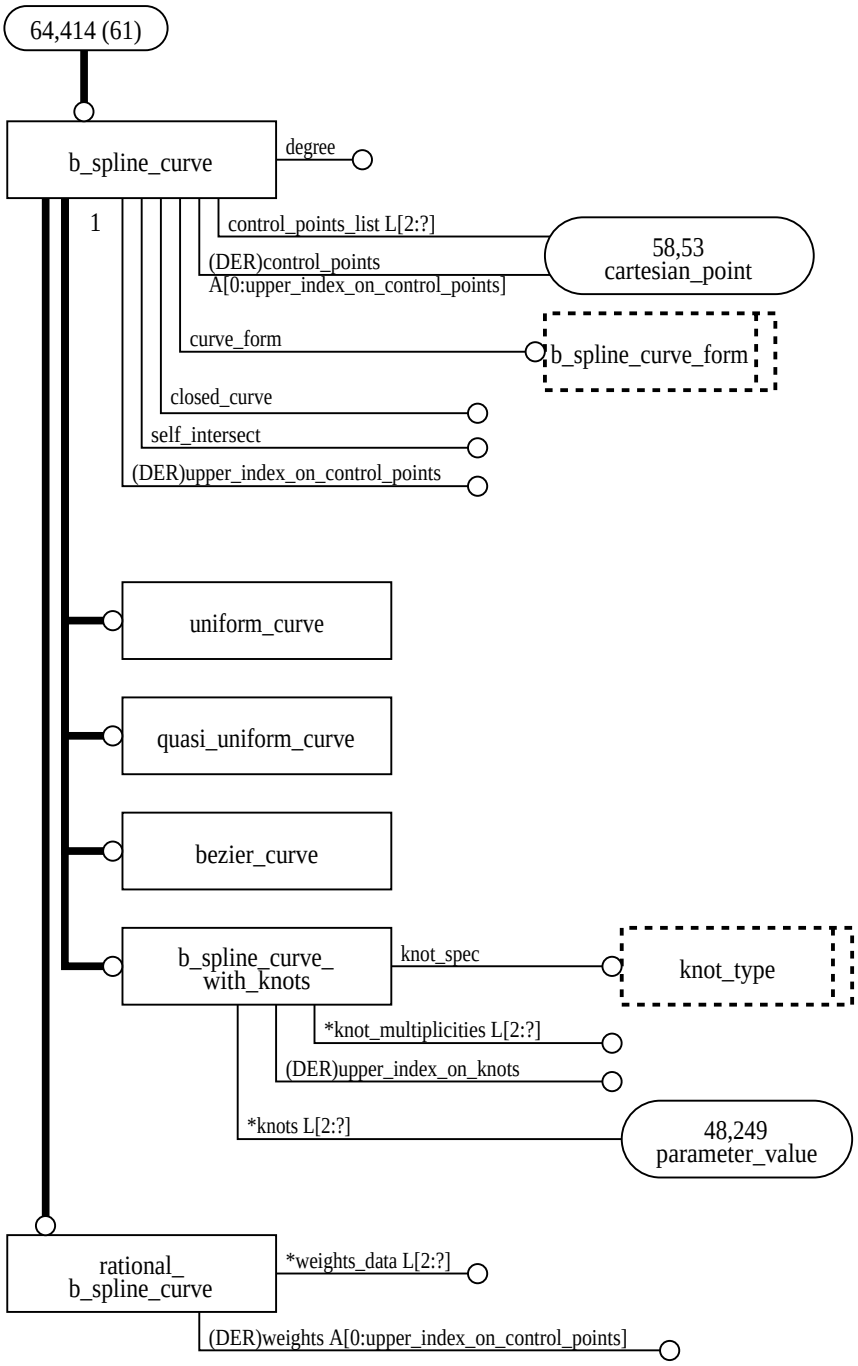


Figure H.64 – AIM diagram in EXPRESS_G: 64 of 124

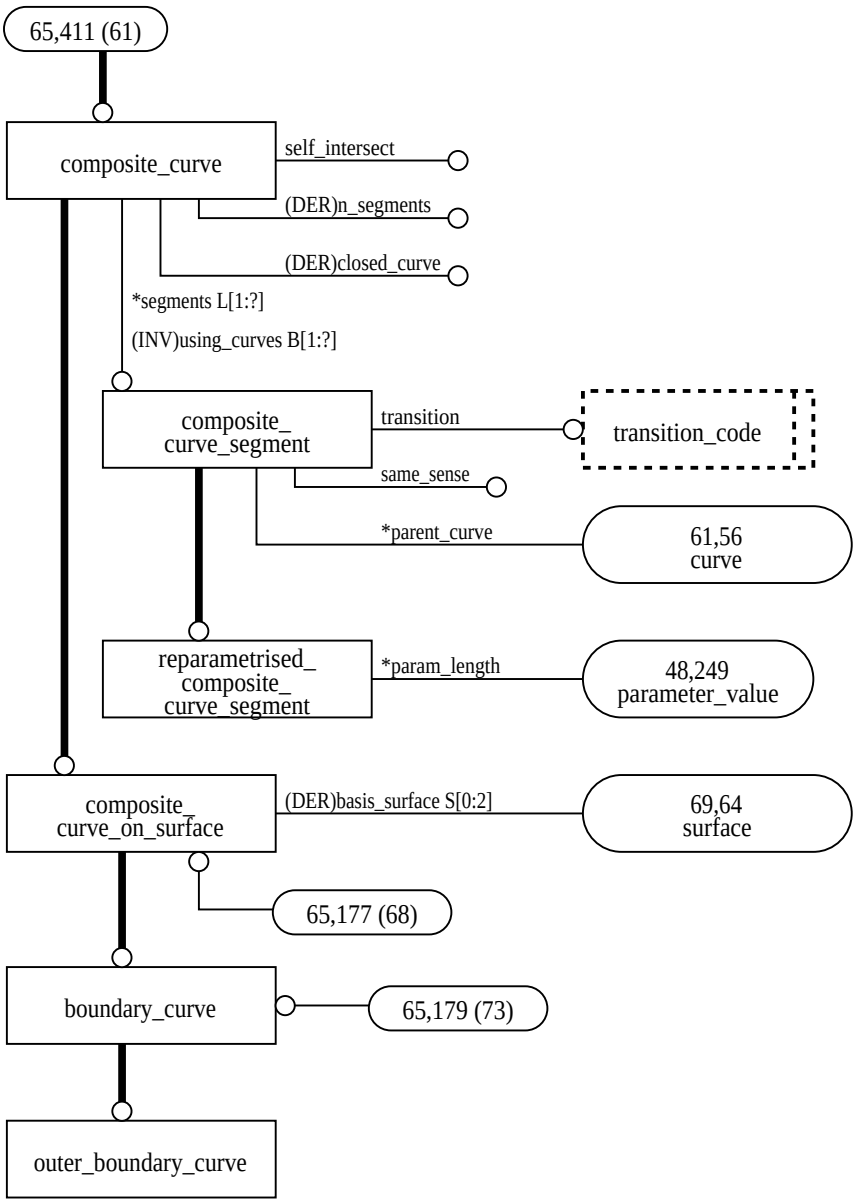


Figure H.65 – AIM diagram in EXPRESS_G: 65 of 124

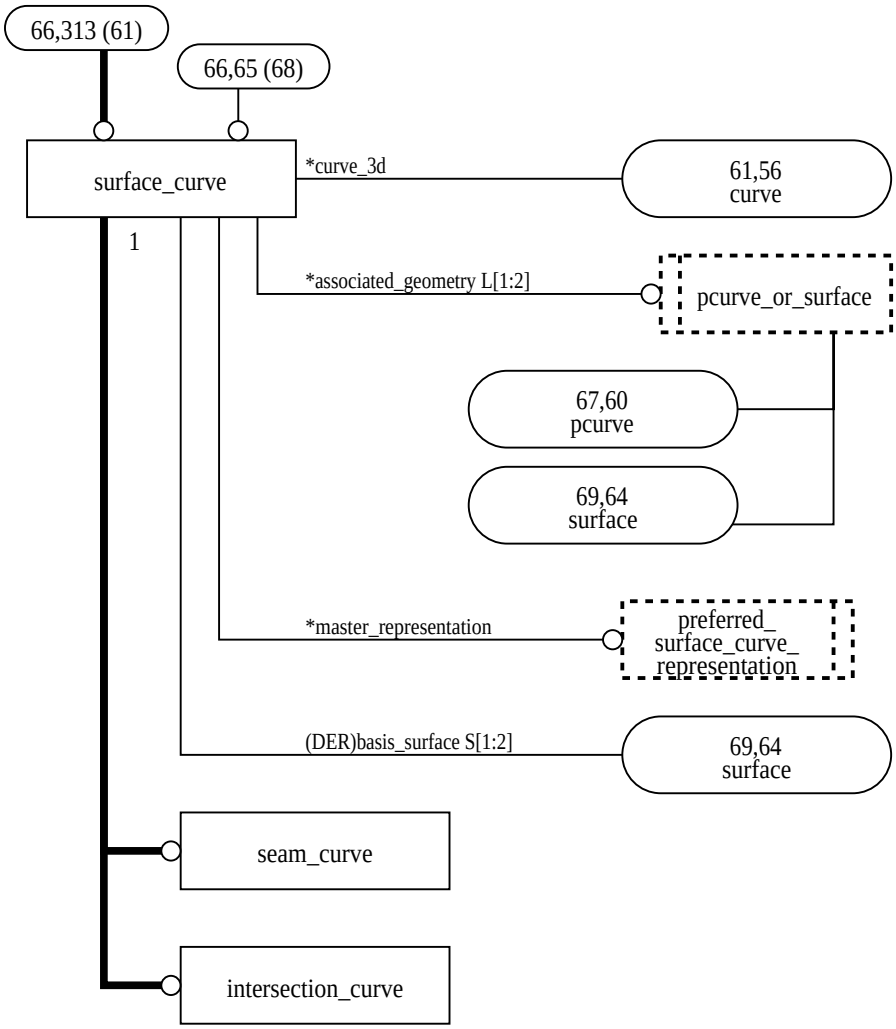


Figure H.66 – AIM diagram in EXPRESS_G: 66 of 124

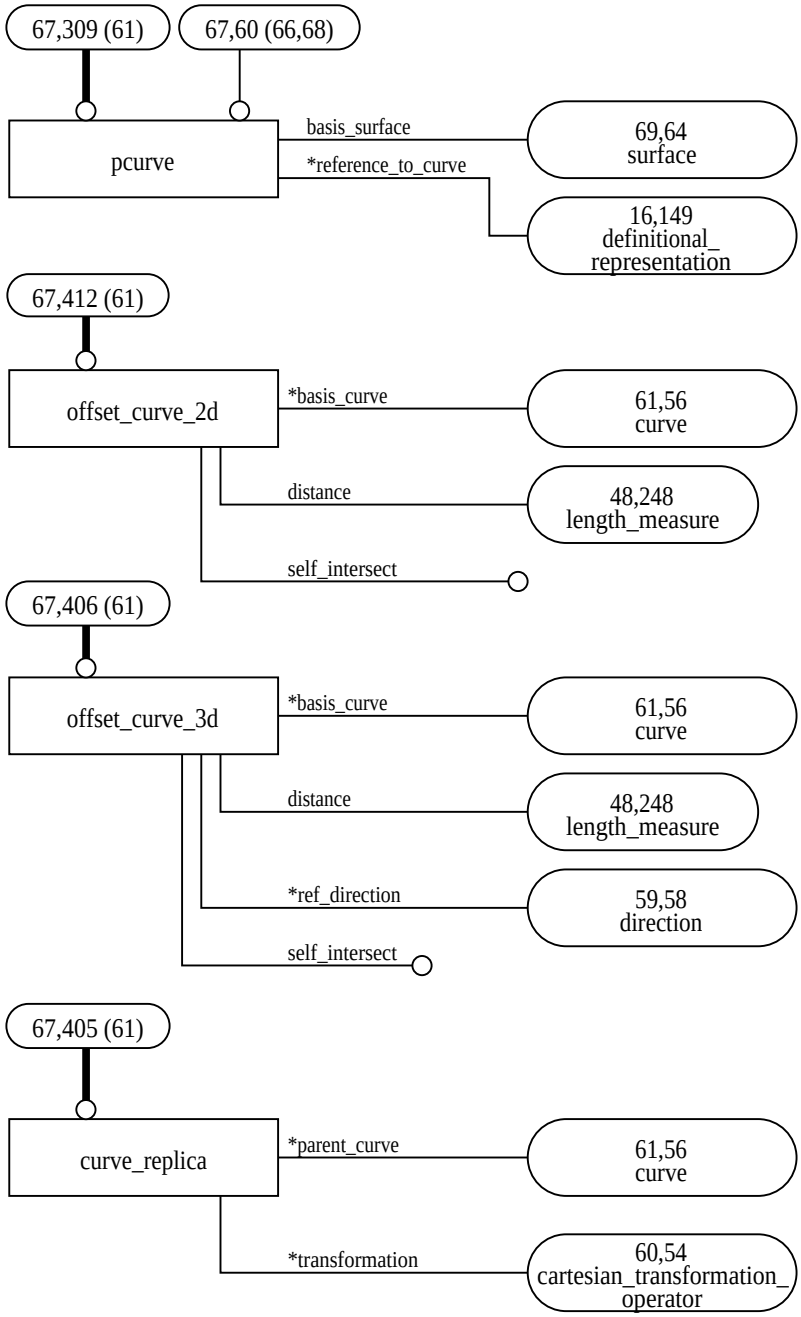


Figure H.67 – AIM diagram in EXPRESS_G: 67 of 124

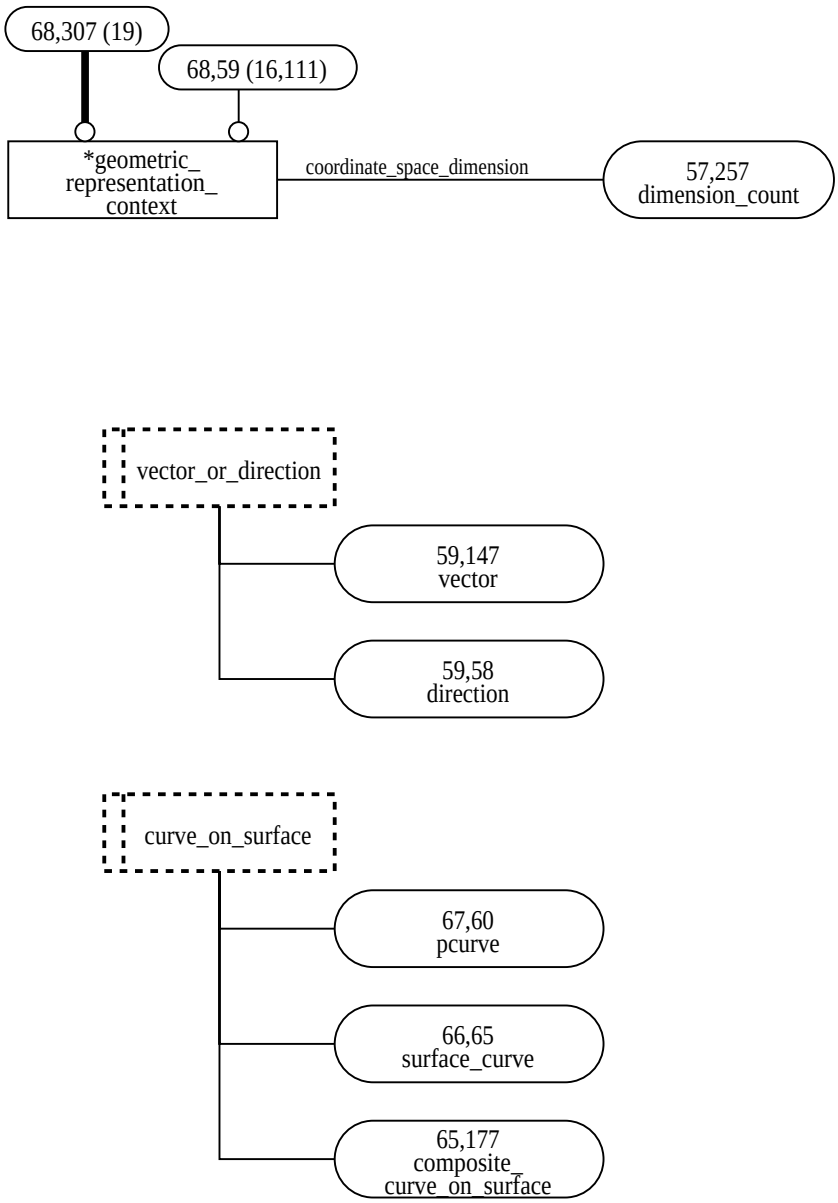


Figure H.68 – AIM diagram in EXPRESS_G: 68 of 124

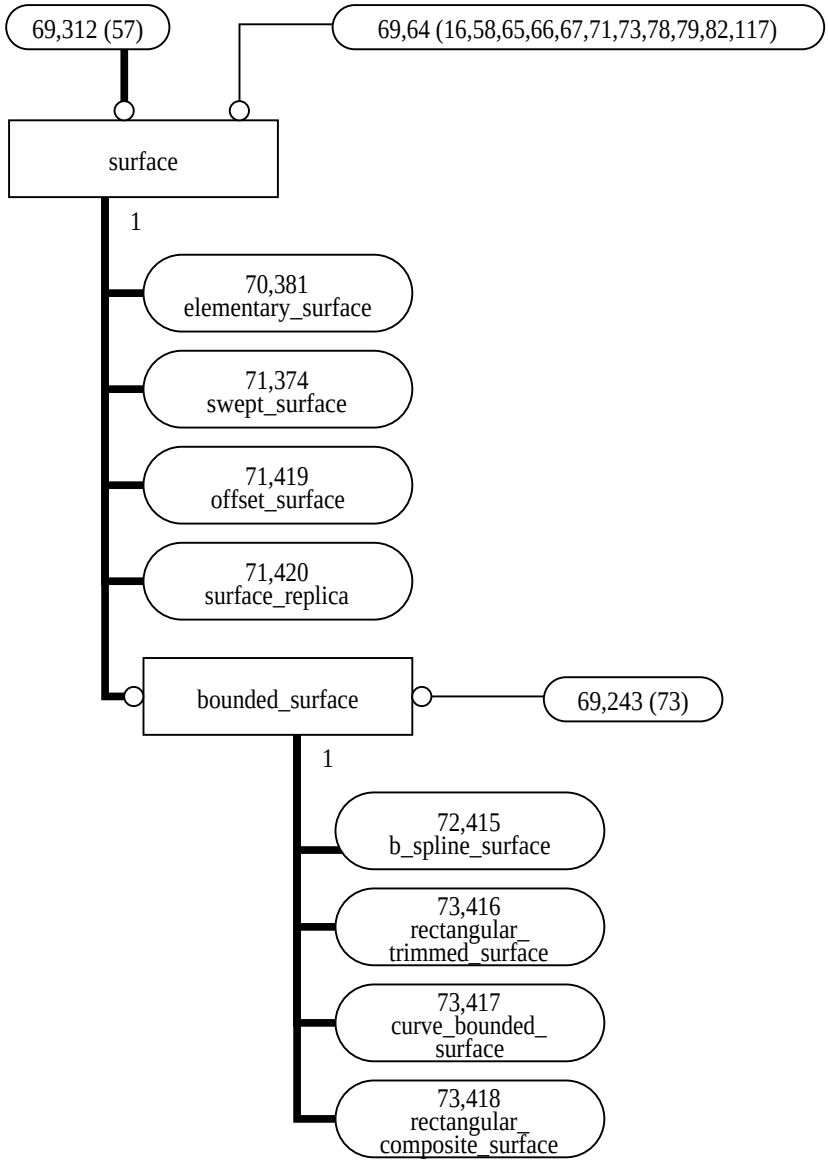


Figure H.69 – AIM diagram in EXPRESS_G: 69 of 124

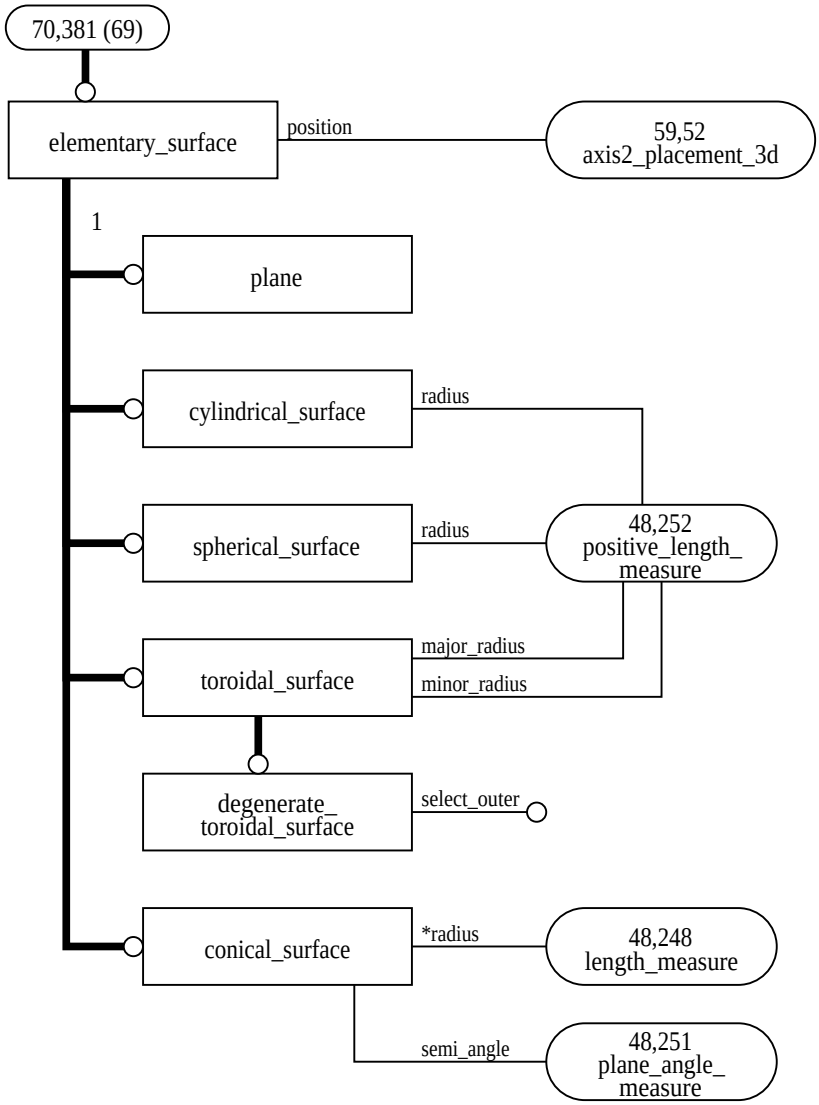


Figure H.70 – AIM diagram in EXPRESS_G: 70 of 124

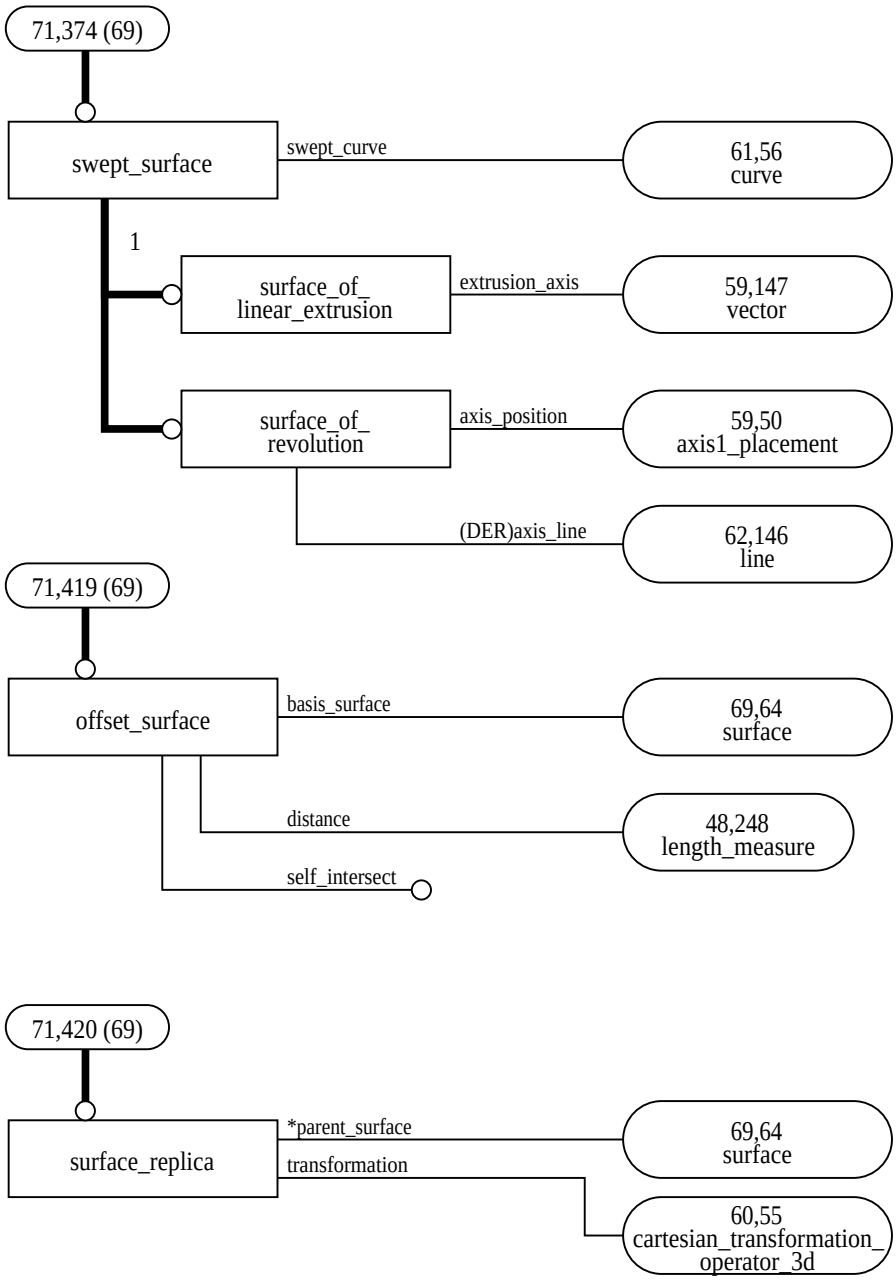


Figure H.71 – AIM diagram in EXPRESS_G: 71 of 124

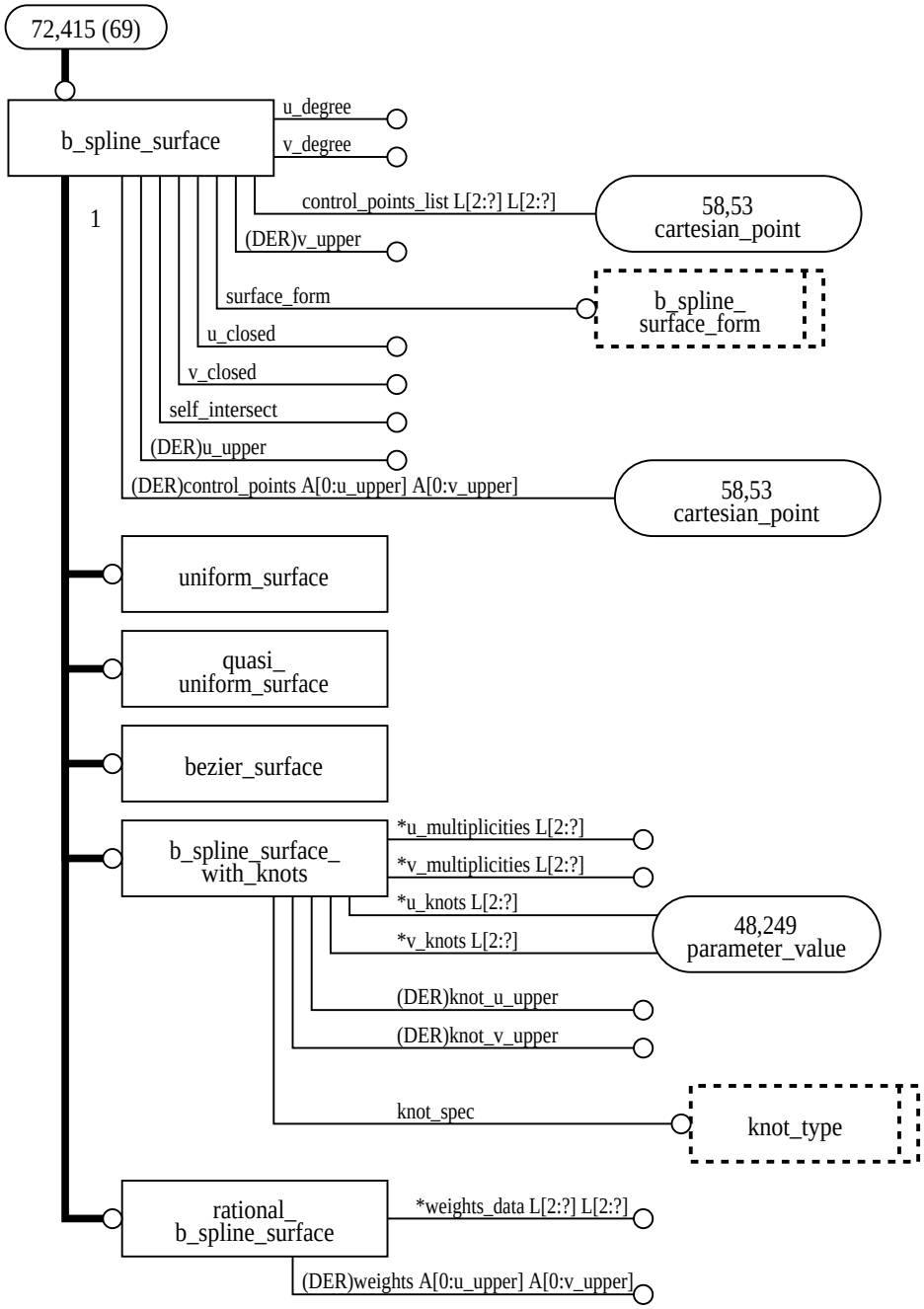


Figure H.72 – AIM diagram in EXPRESS_G: 72 of 124

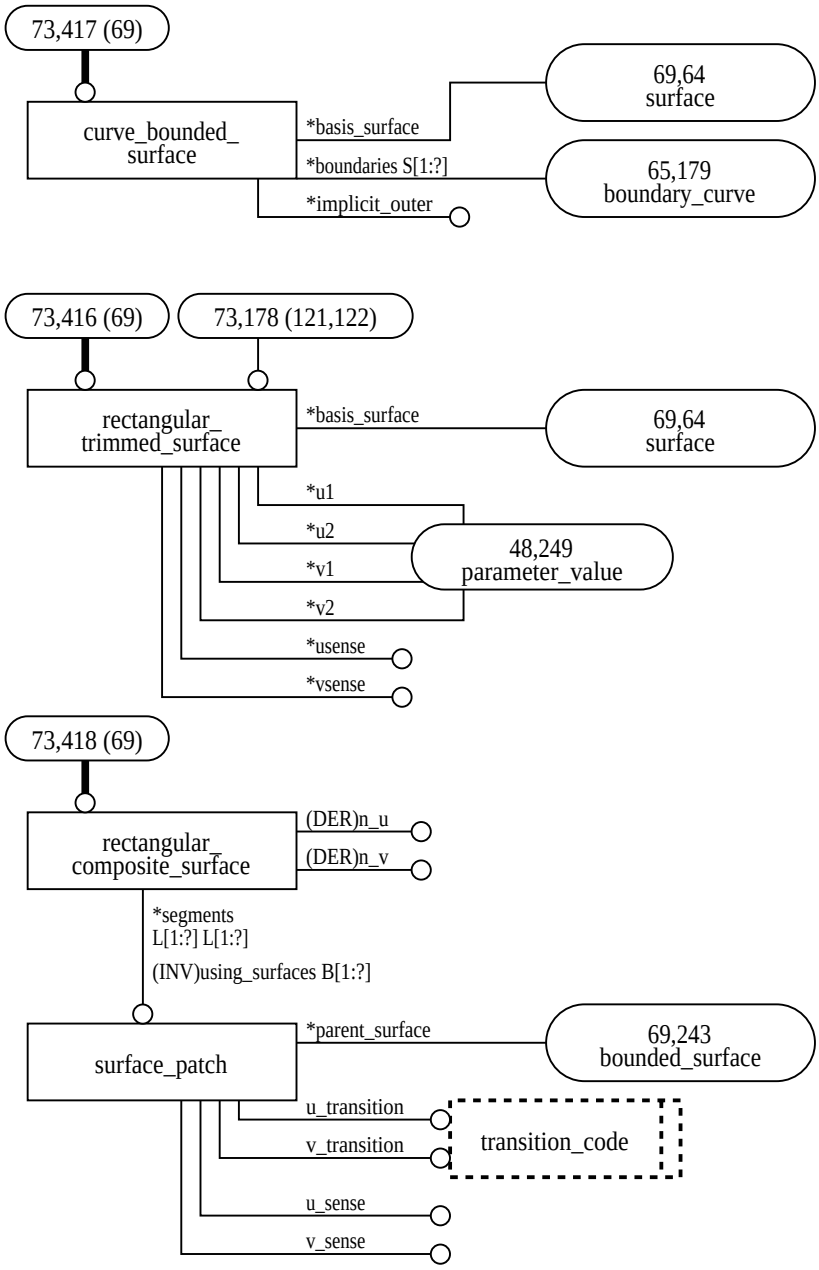


Figure H.73 – AIM diagram in EXPRESS_G: 73 of 124

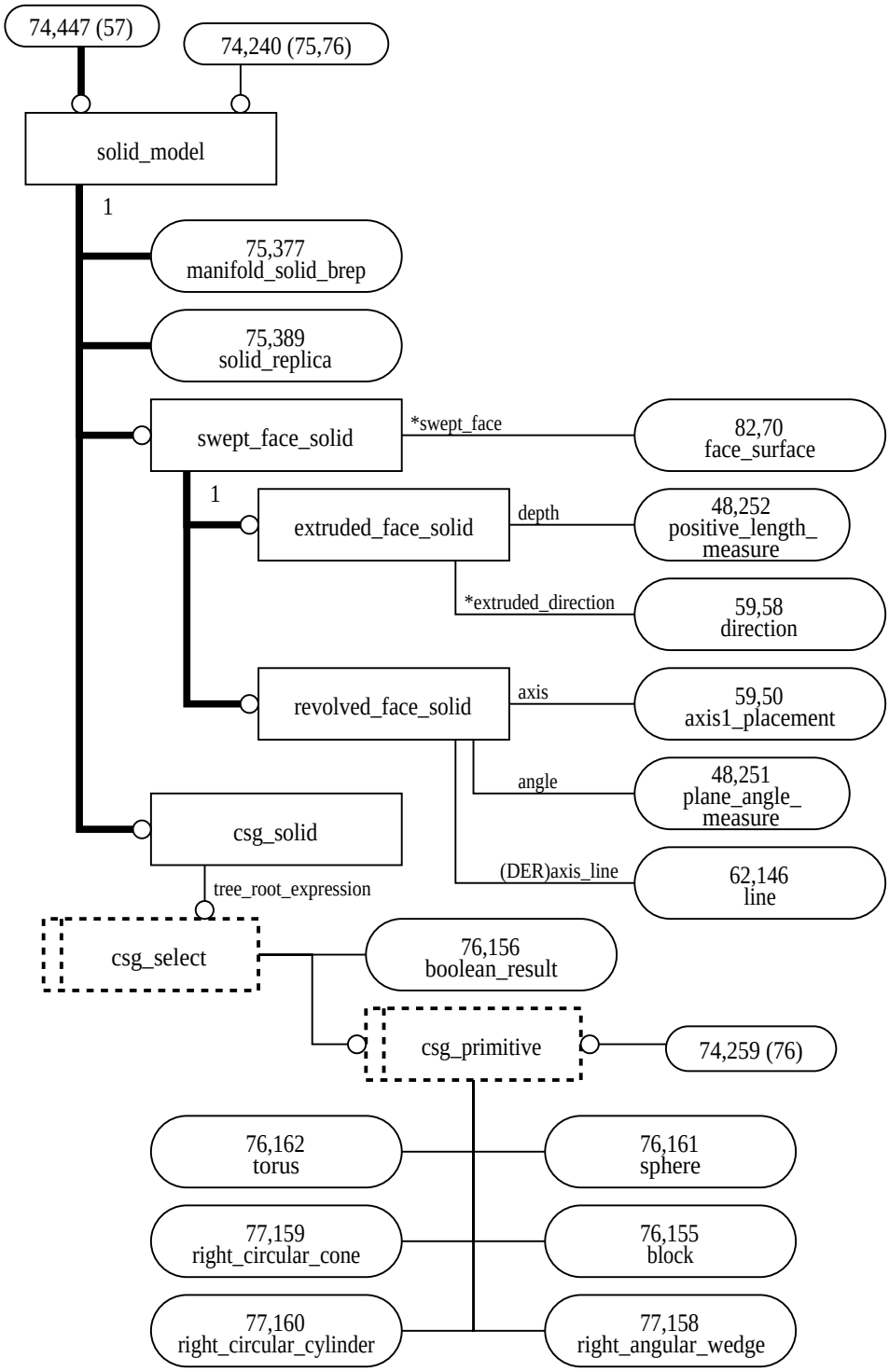
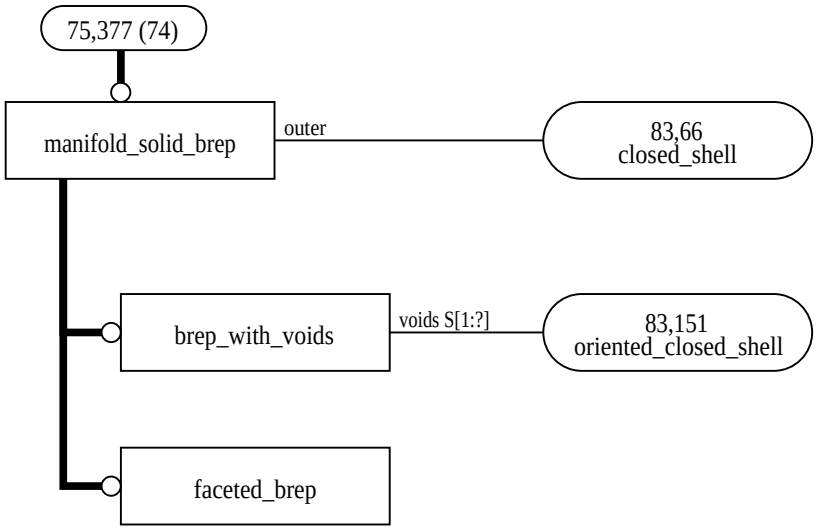


Figure H.74 – AIM diagram in EXPRESS_G: 74 of 124



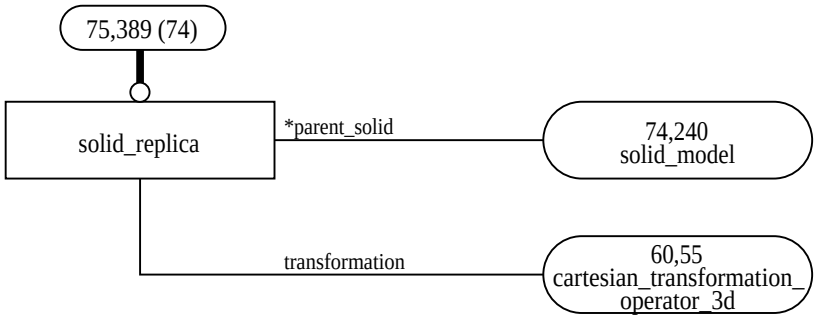


Figure H.75 – AIM diagram in EXPRESS_G: 75 of 124

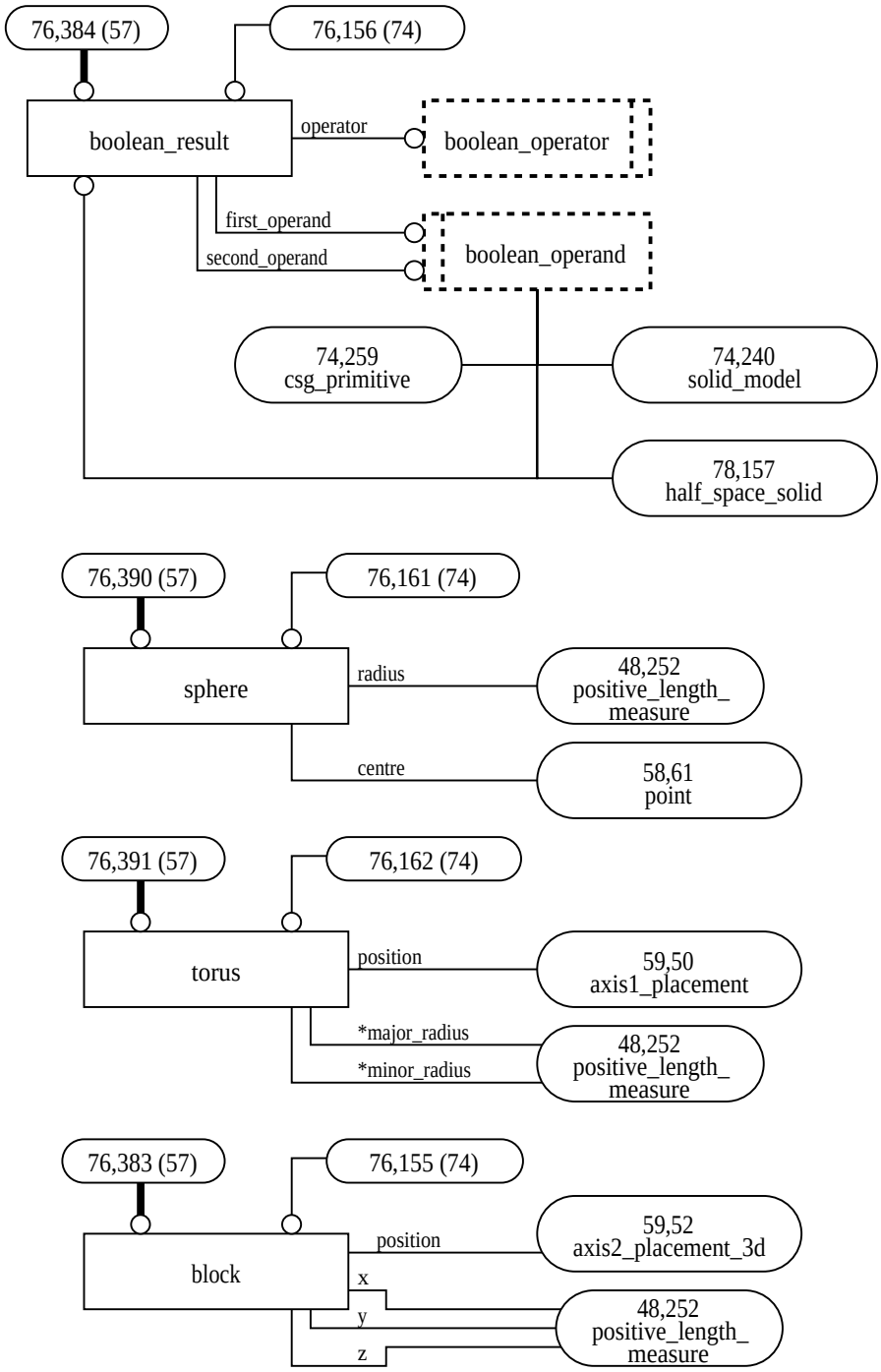


Figure H.76 – AIM diagram in EXPRESS_G: 76 of 124

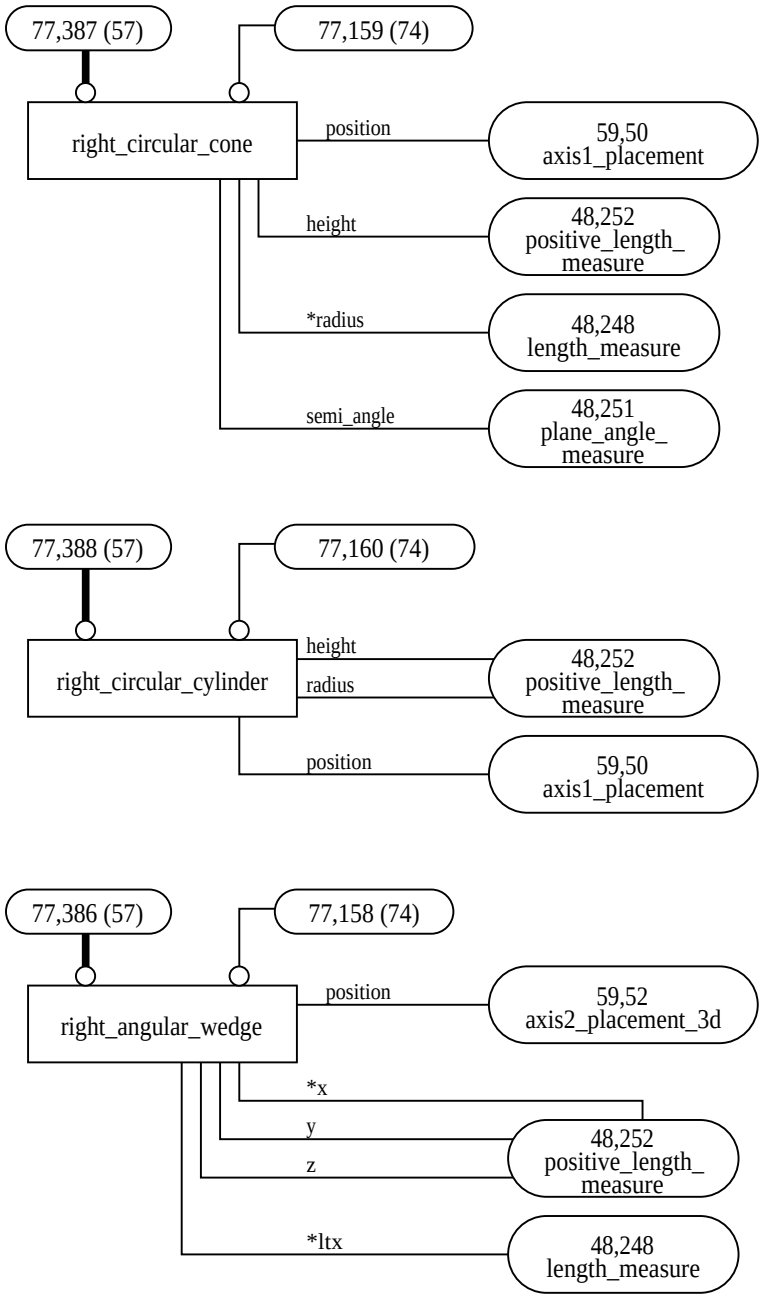


Figure H.77 – AIM diagram in EXPRESS_G: 77 of 124

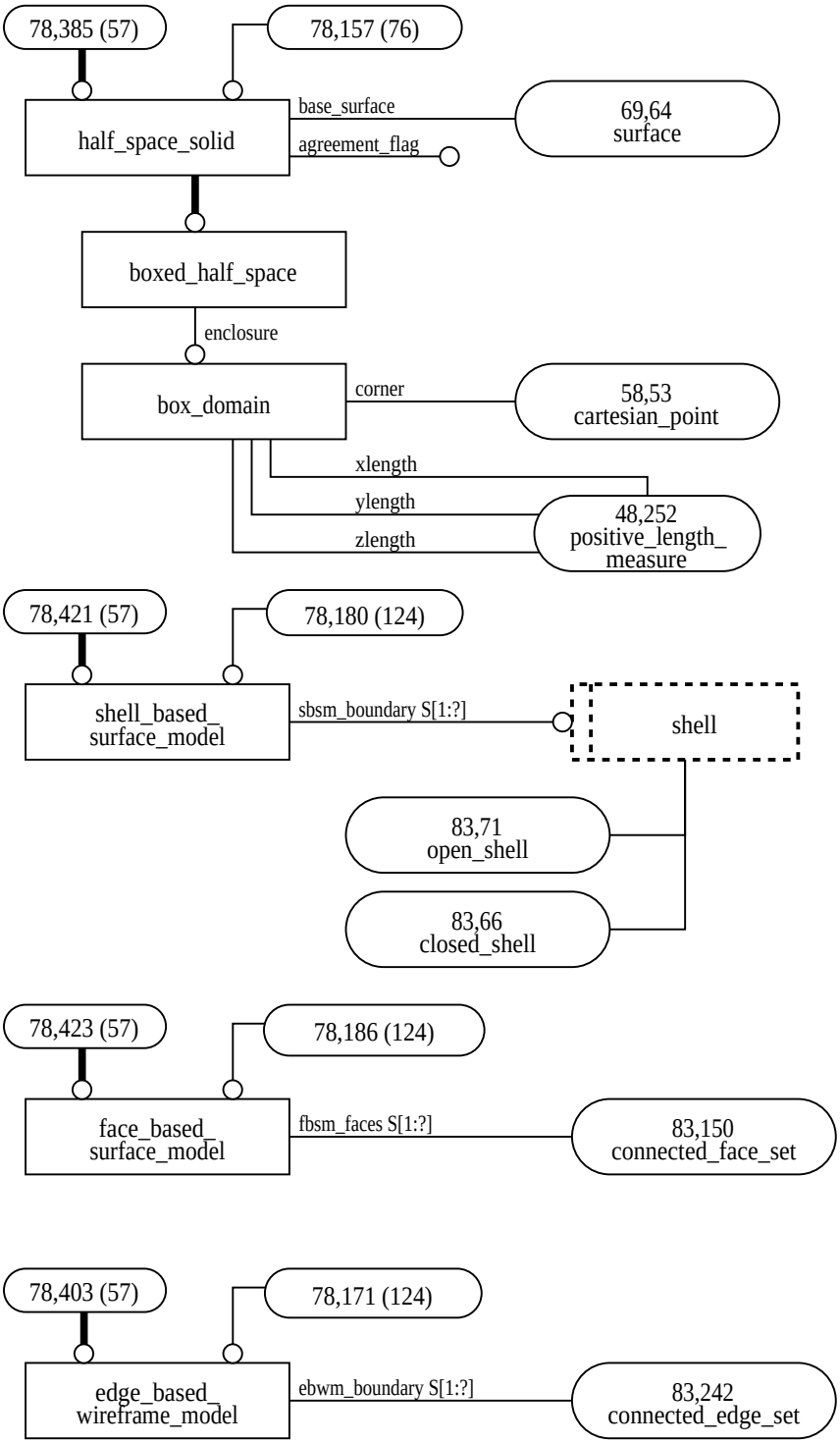


Figure H.78 – AIM diagram in EXPRESS_G: 78 of 124

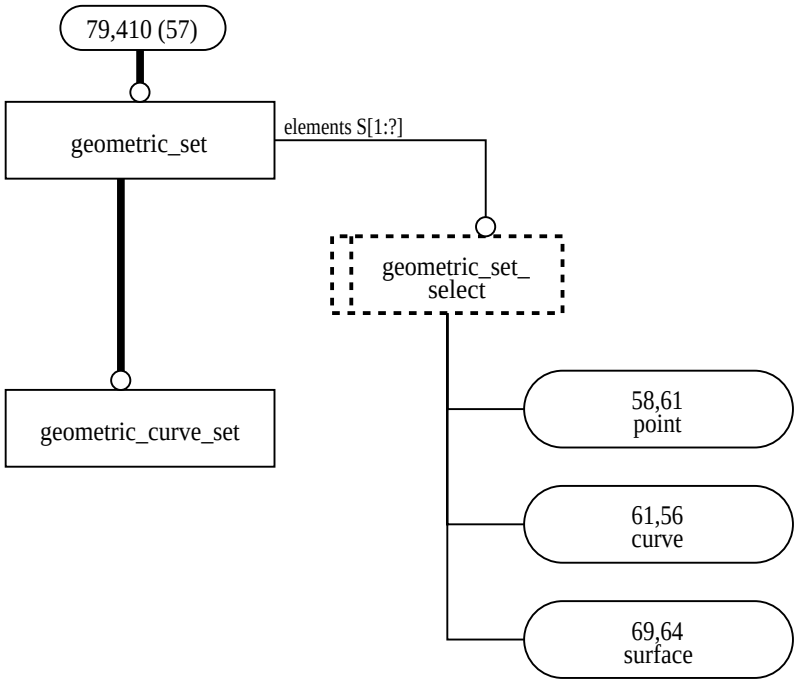


Figure H.79 – AIM diagram in EXPRESS_G: 79 of 124

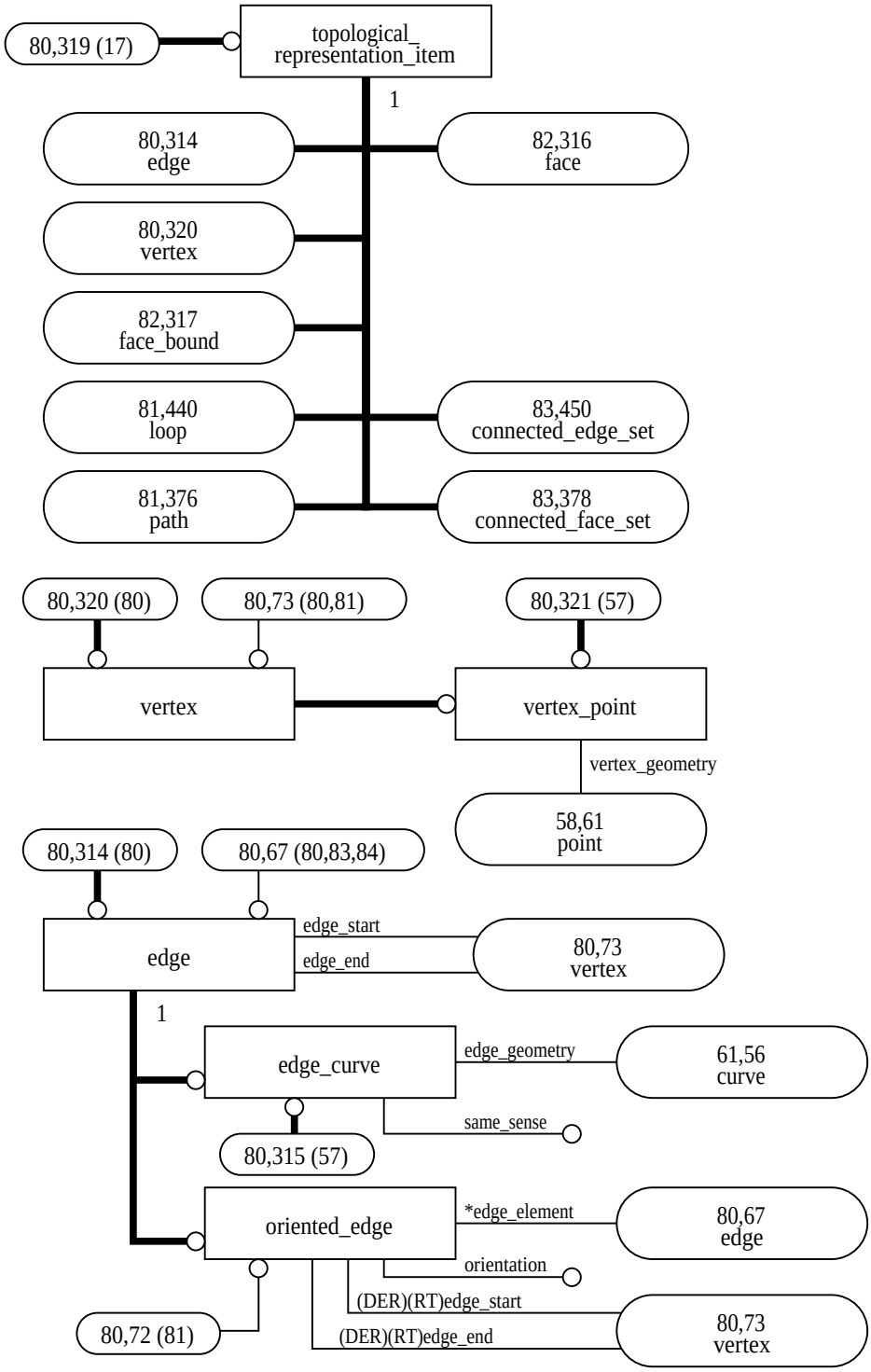


Figure H.80 – AIM diagram in EXPRESS_G: 80 of 124

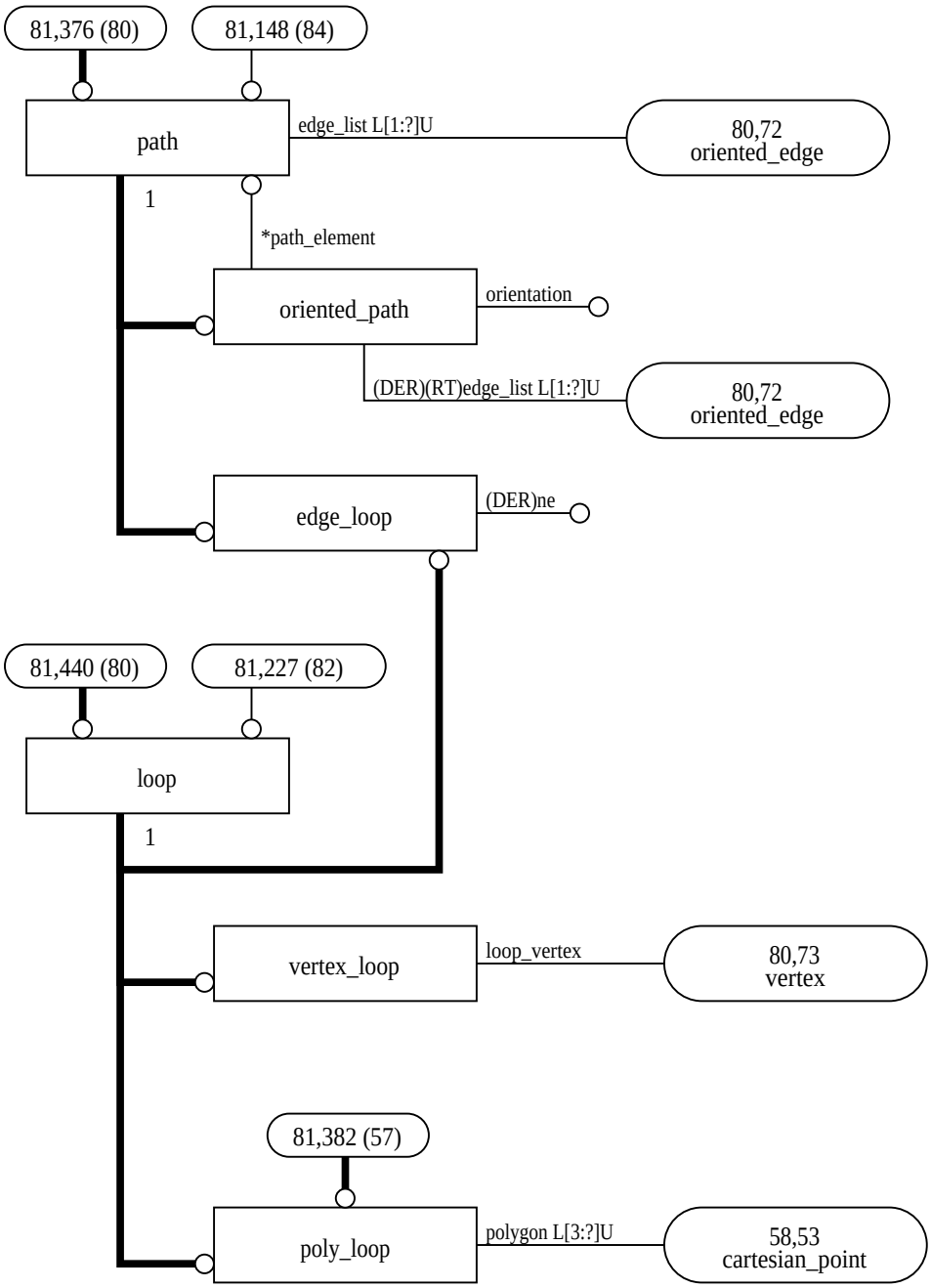


Figure H.81 – AIM diagram in EXPRESS_G: 81 of 124

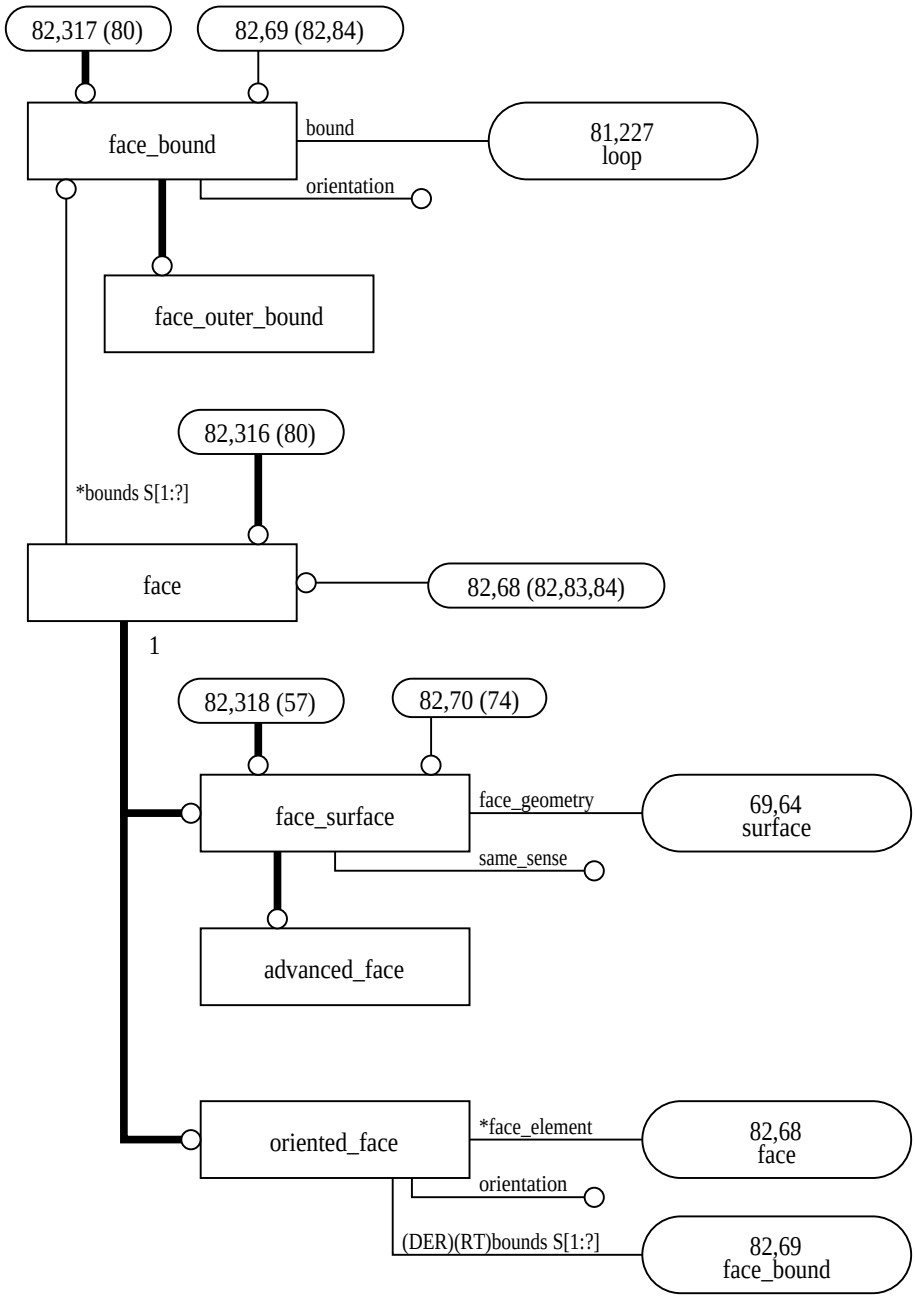


Figure H.82 – AIM diagram in EXPRESS_G: 82 of 124

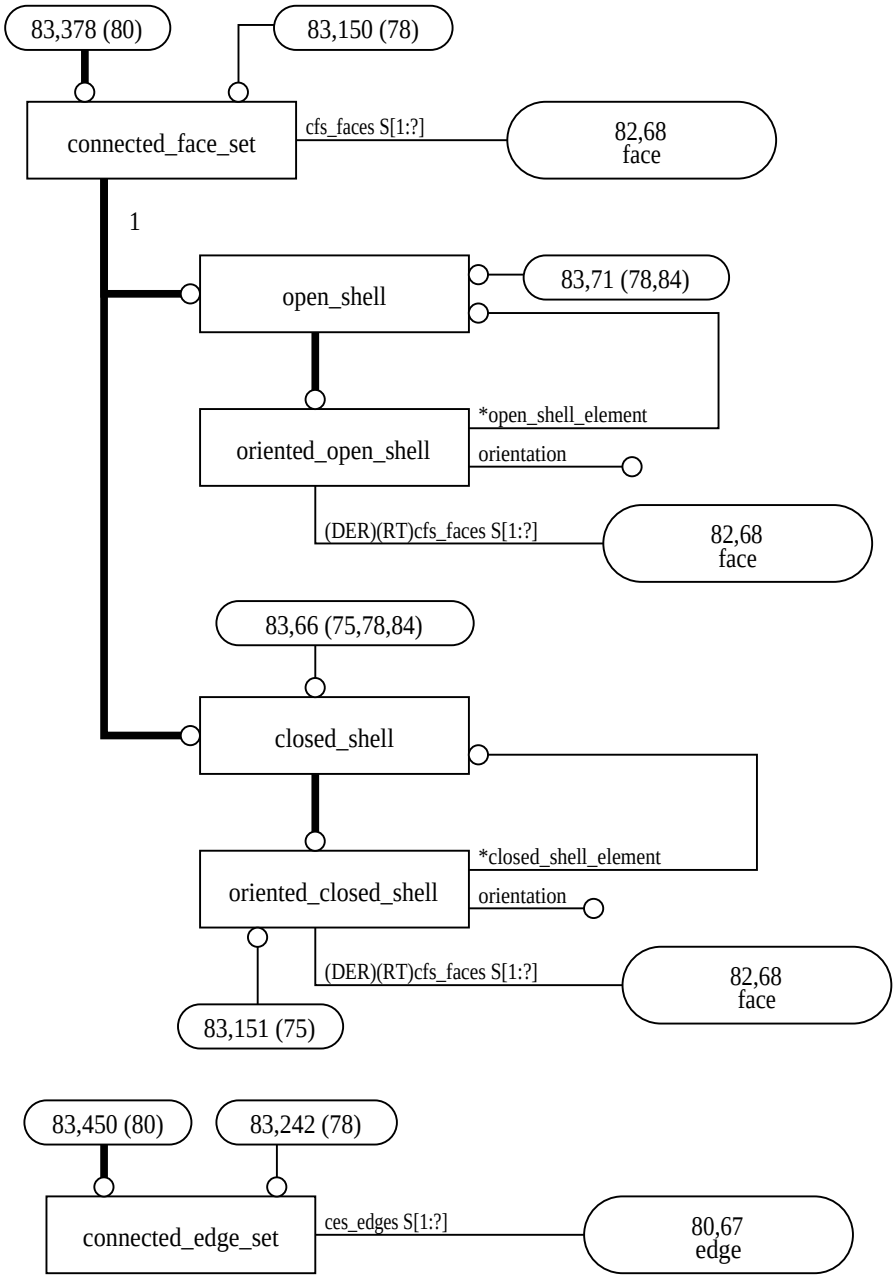


Figure H.83 – AIM diagram in EXPRESS_G: 83 of 124

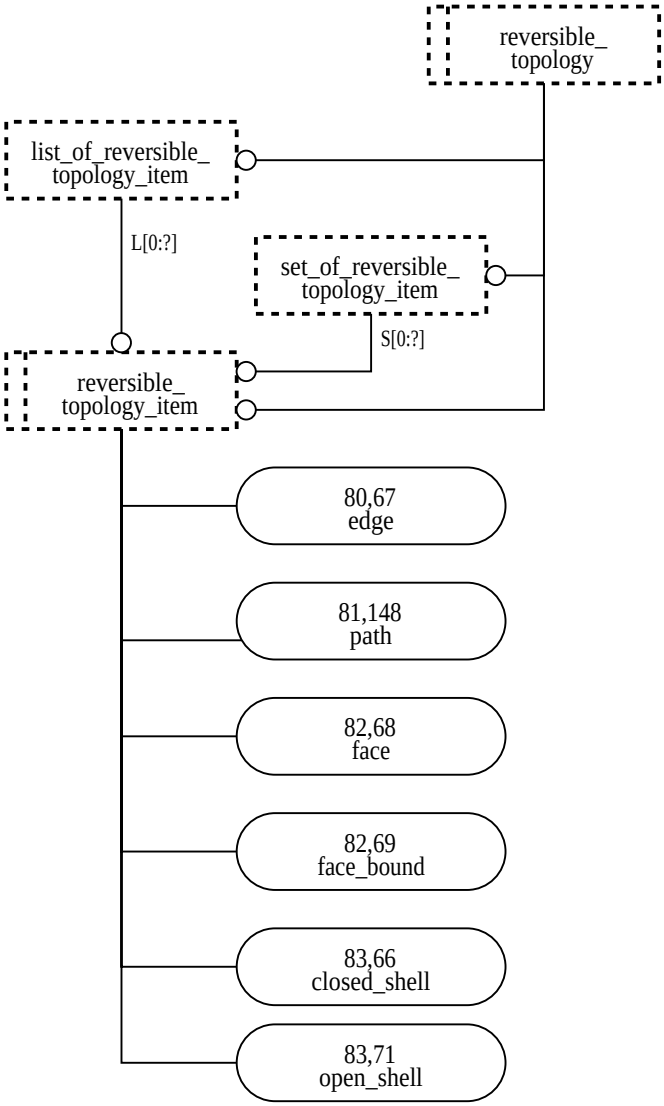


Figure H.84 – AIM diagram in EXPRESS_G: 84 of 124

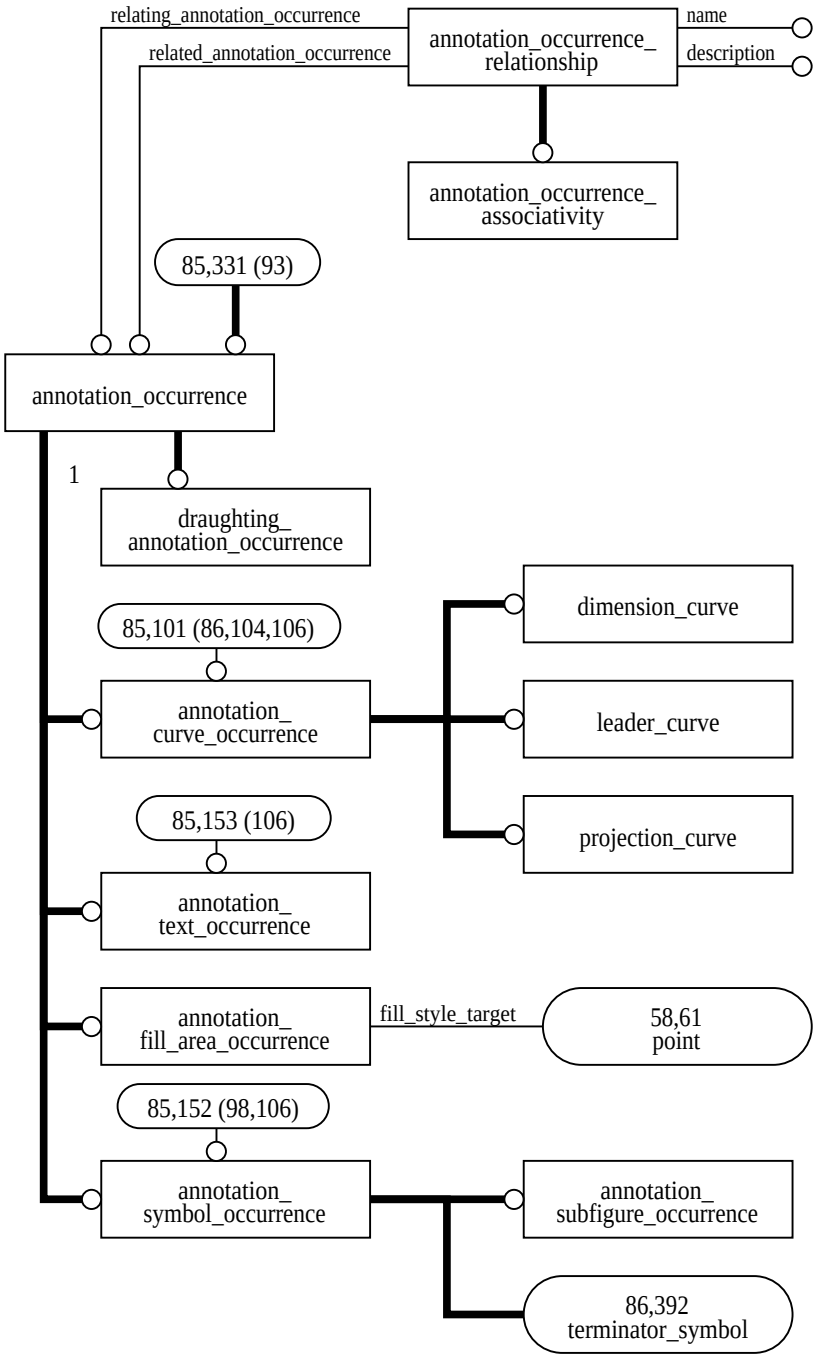


Figure H.85 – AIM diagram in EXPRESS_G: 85 of 124

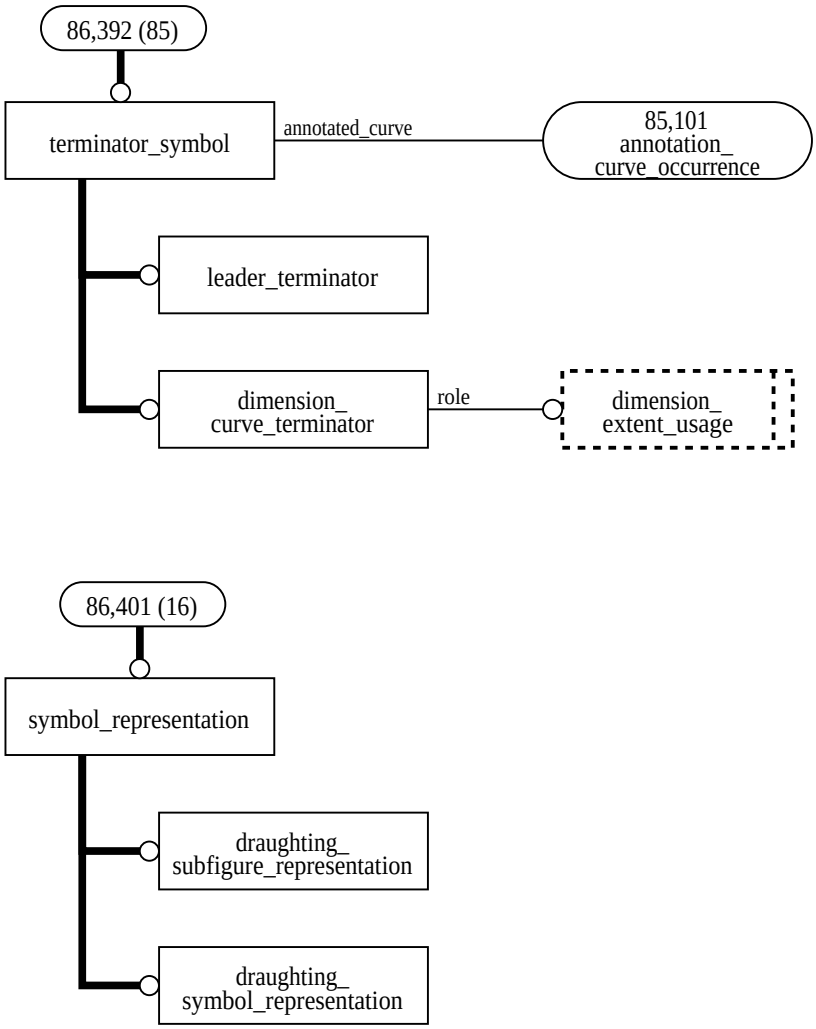


Figure H.86 – AIM diagram in EXPRESS_G: 86 of 124

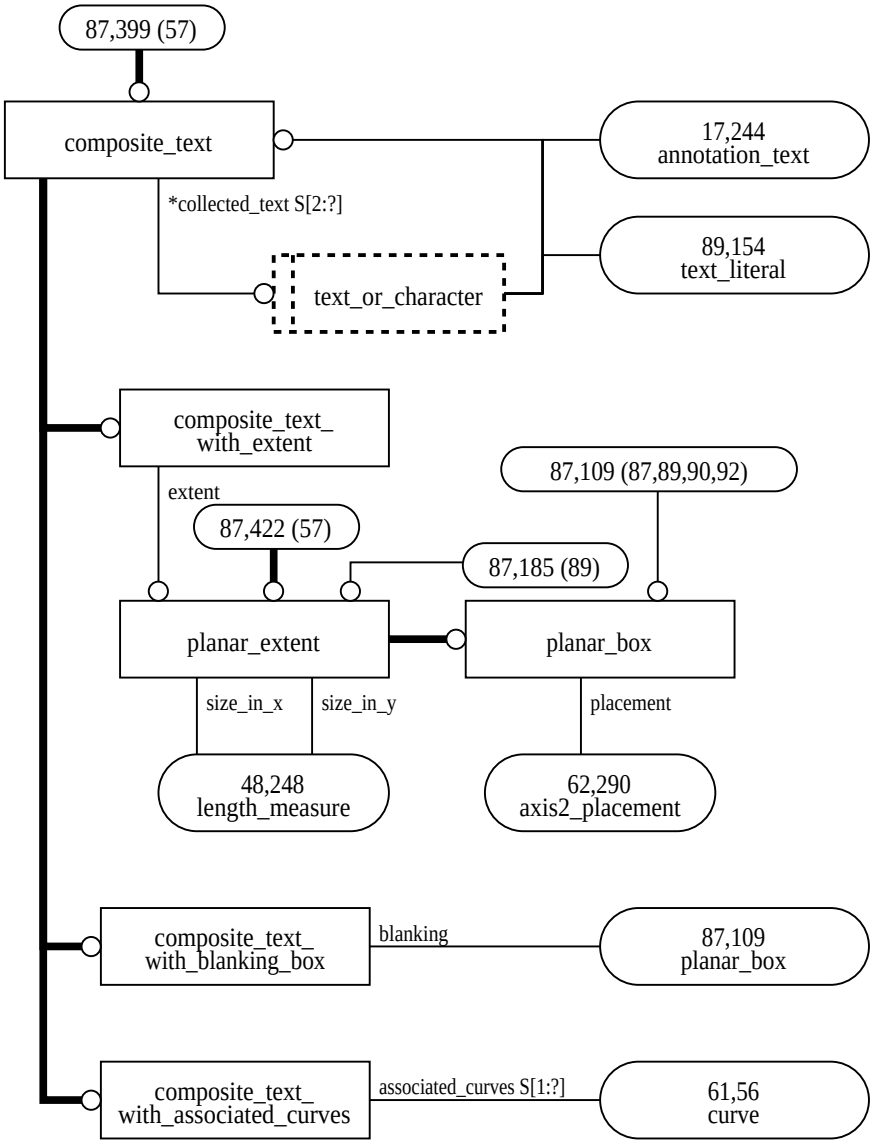


Figure H.87 – AIM diagram in EXPRESS_G: 87 of 124

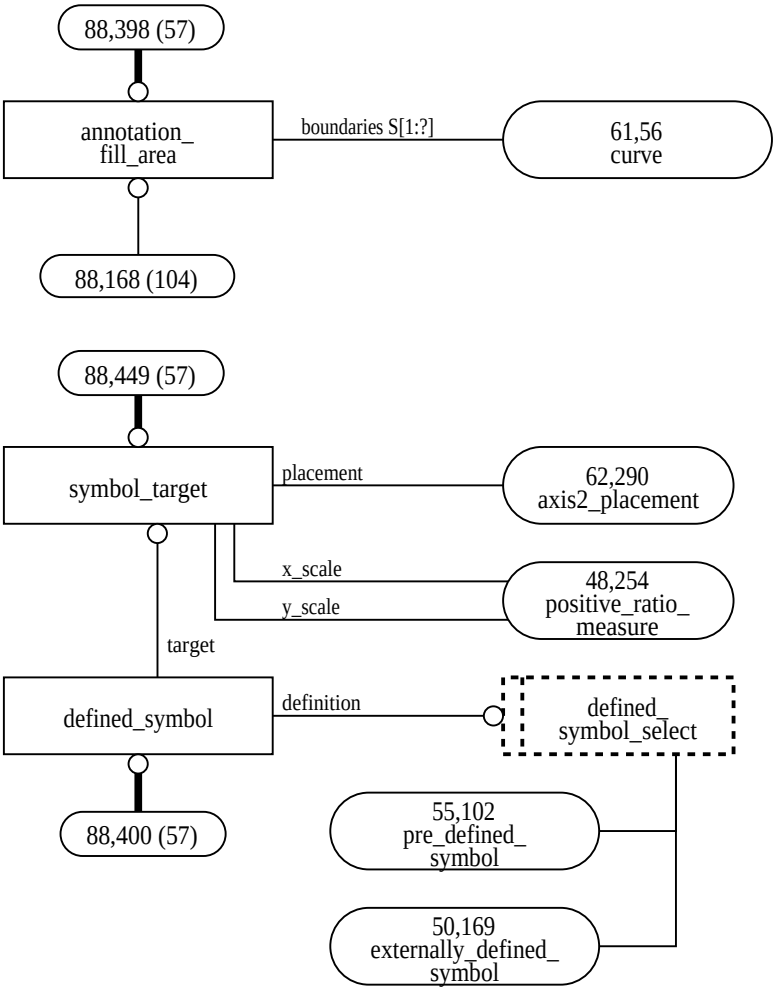


Figure H.88 – AIM diagram in EXPRESS_G: 88 of 124

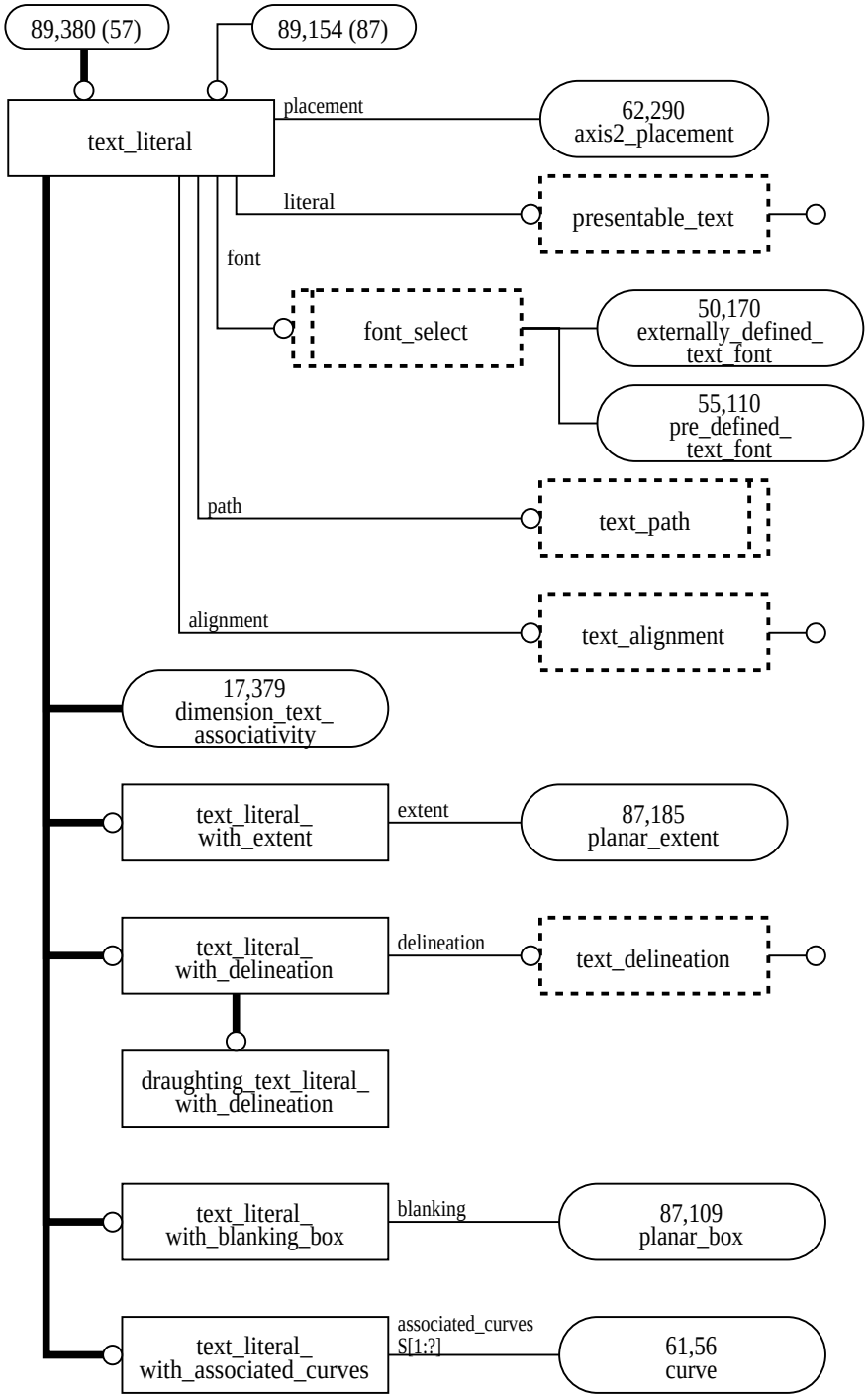


Figure H.89 – AIM diagram in EXPRESS_G: 89 of 124

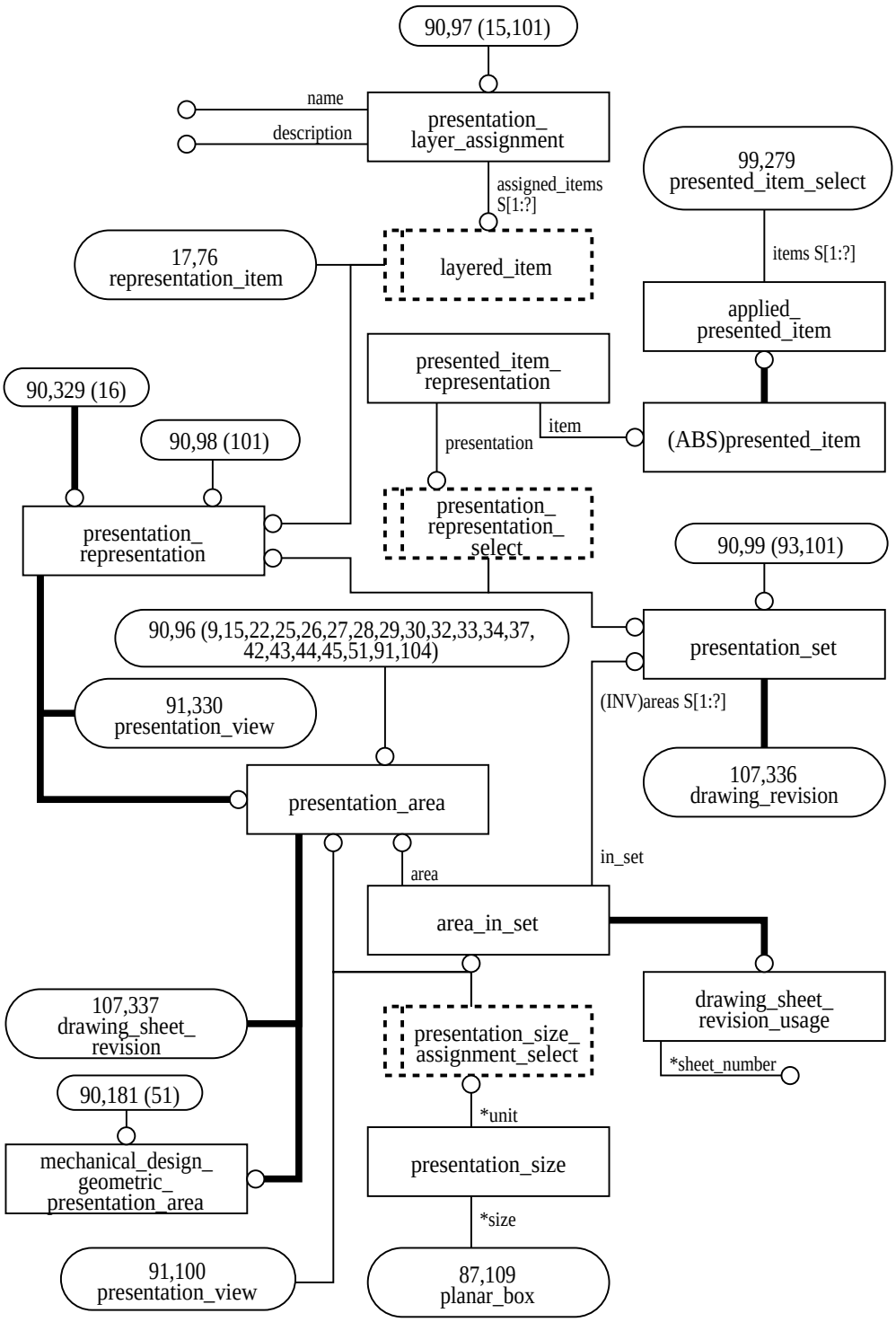


Figure H.90 – AIM diagram in EXPRESS_G: 90 of 124

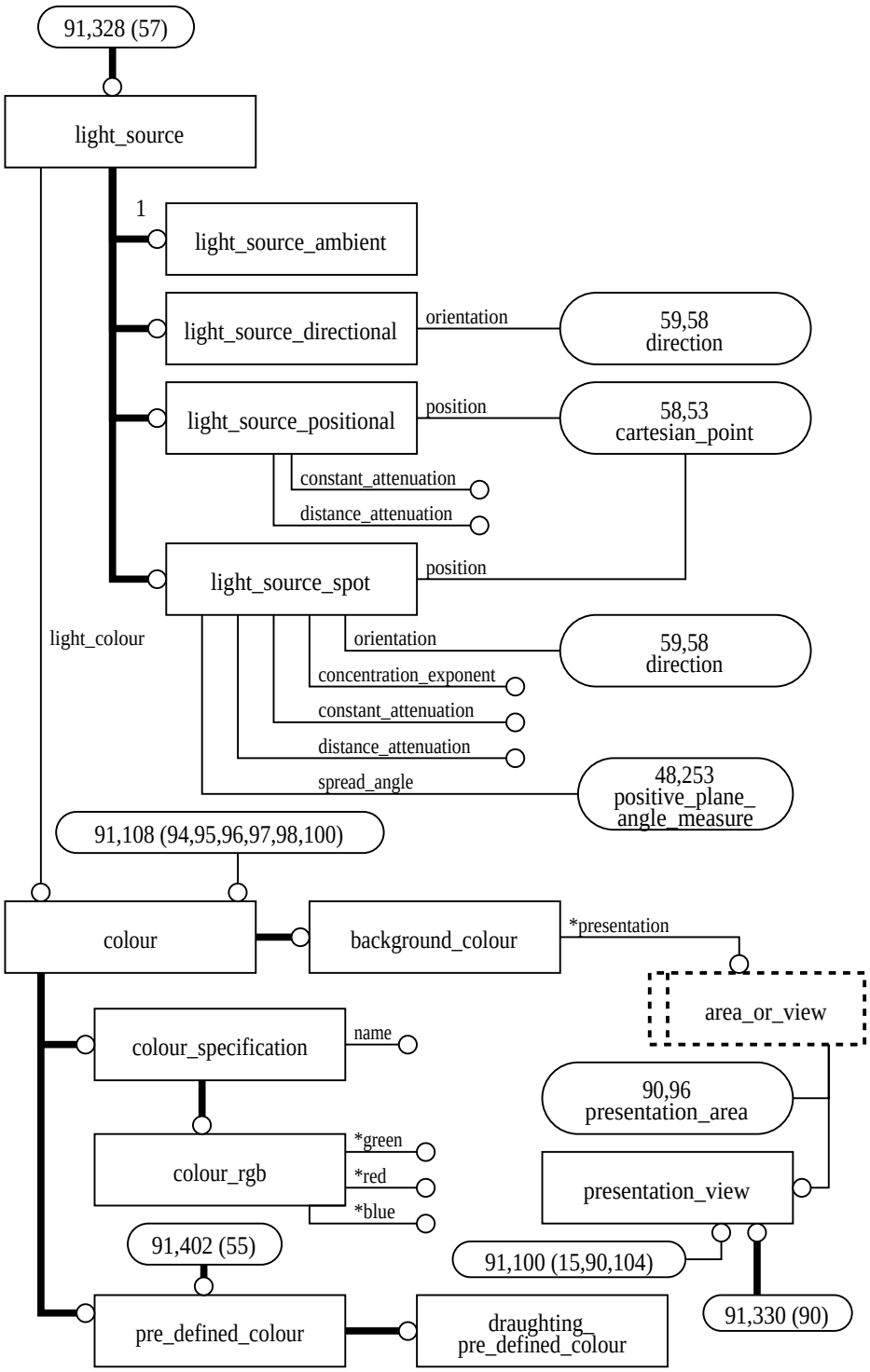


Figure H.91 – AIM diagram in EXPRESS_G: 91 of 124

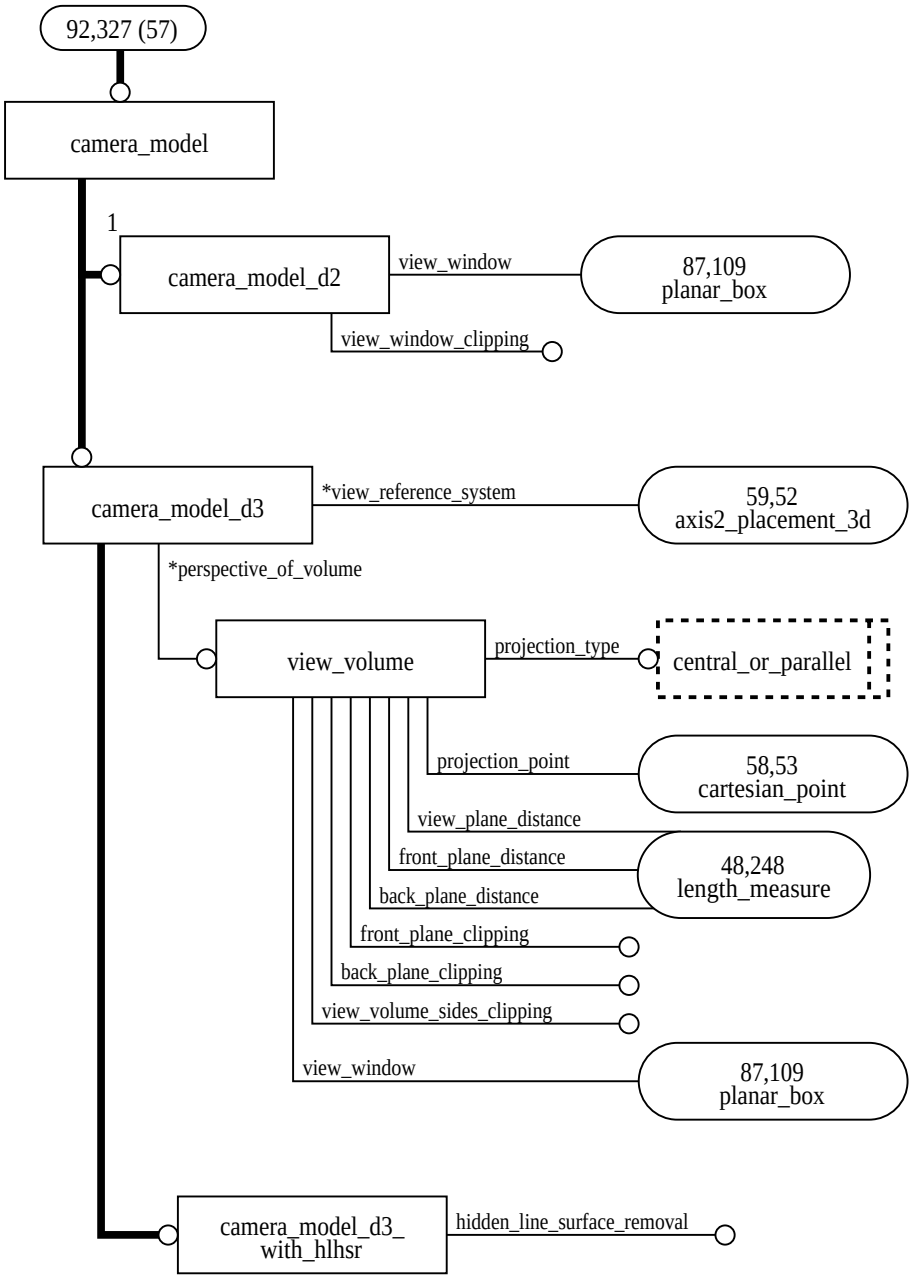


Figure H.92 – AIM diagram in EXPRESS_G: 92 of 124

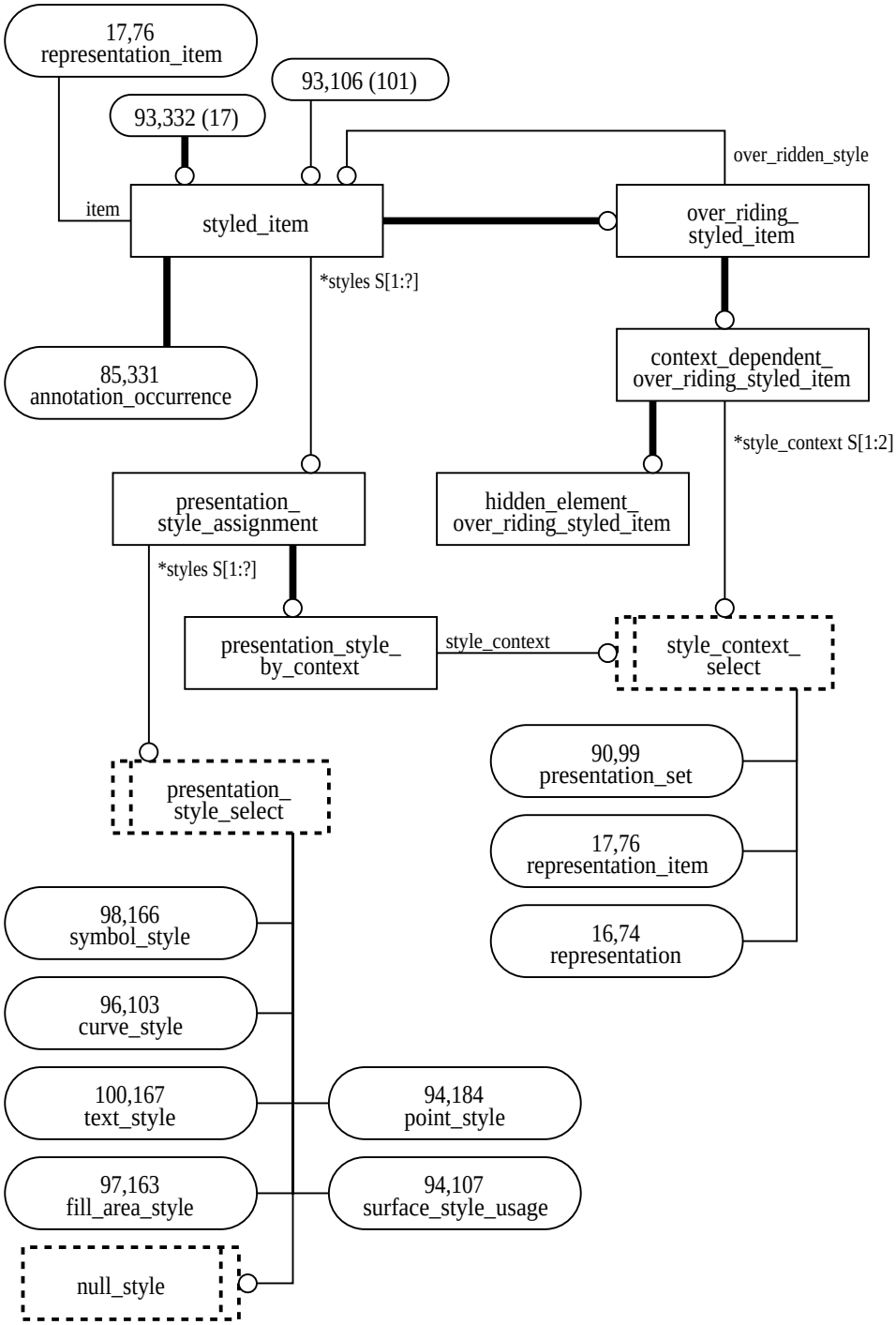


Figure H.93 – AIM diagram in EXPRESS_G: 93 of 124

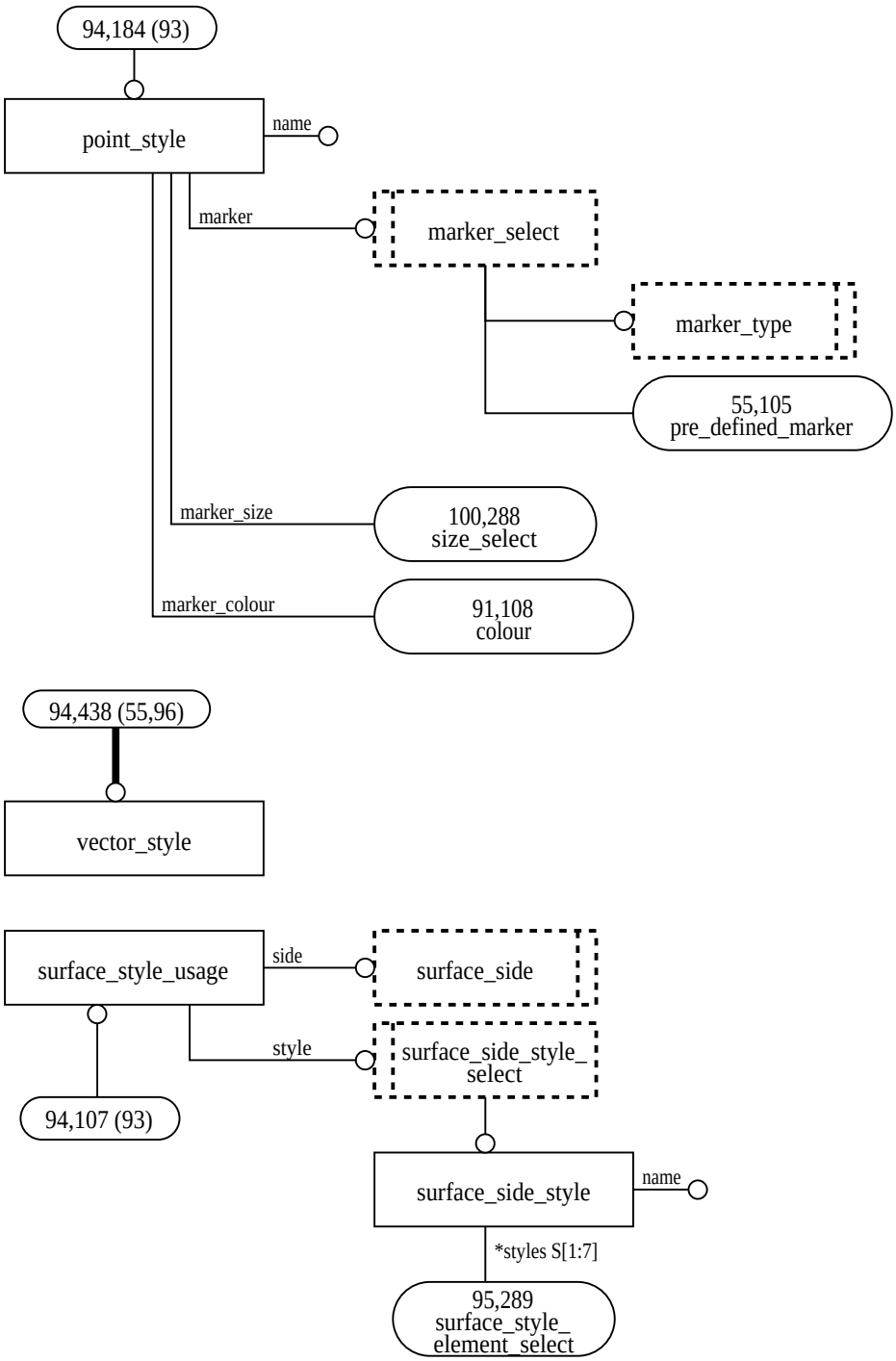


Figure H.94 – AIM diagram in EXPRESS_G: 94 of 124

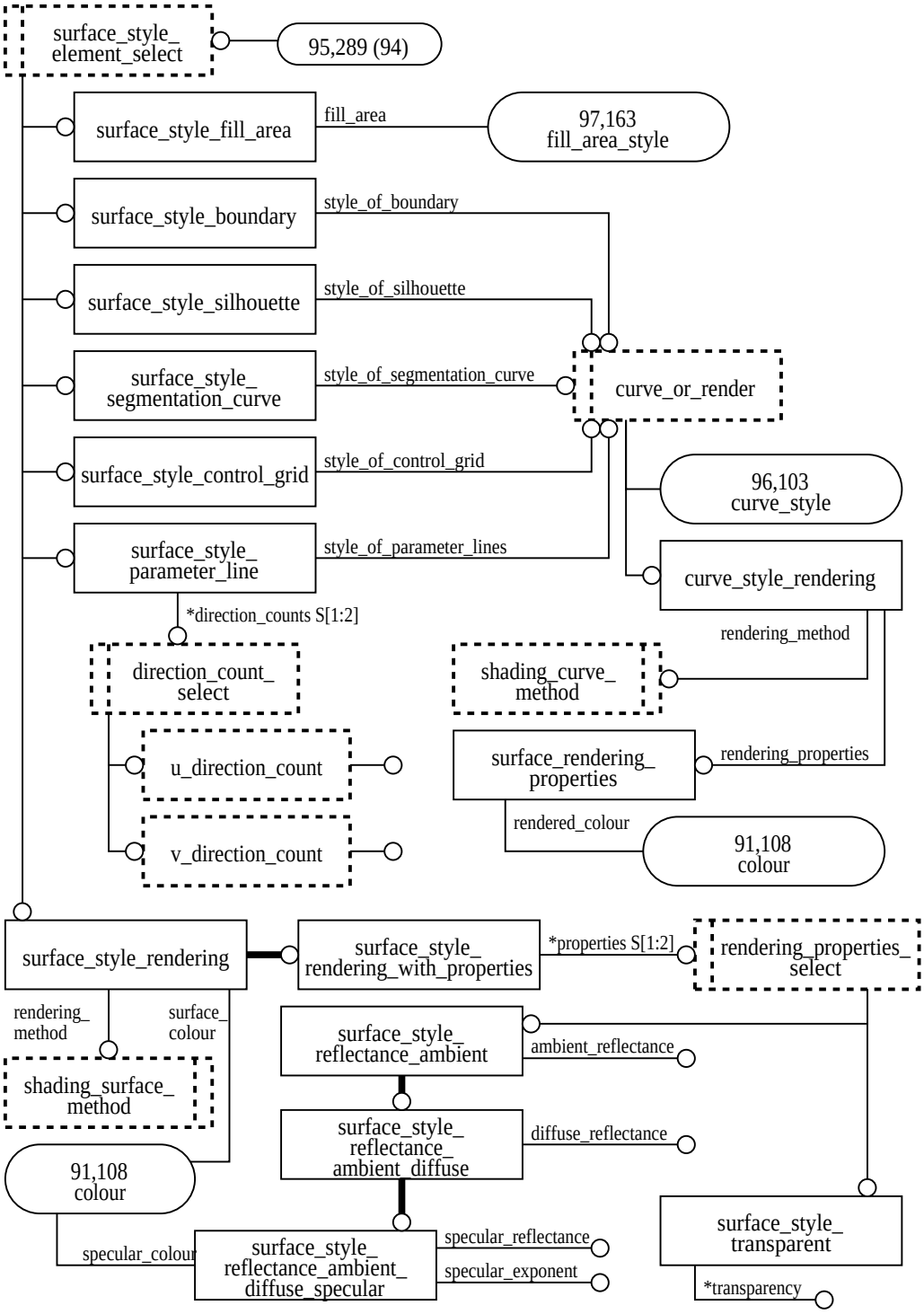


Figure H.95 – AIM diagram in EXPRESS_G: 95 of 124

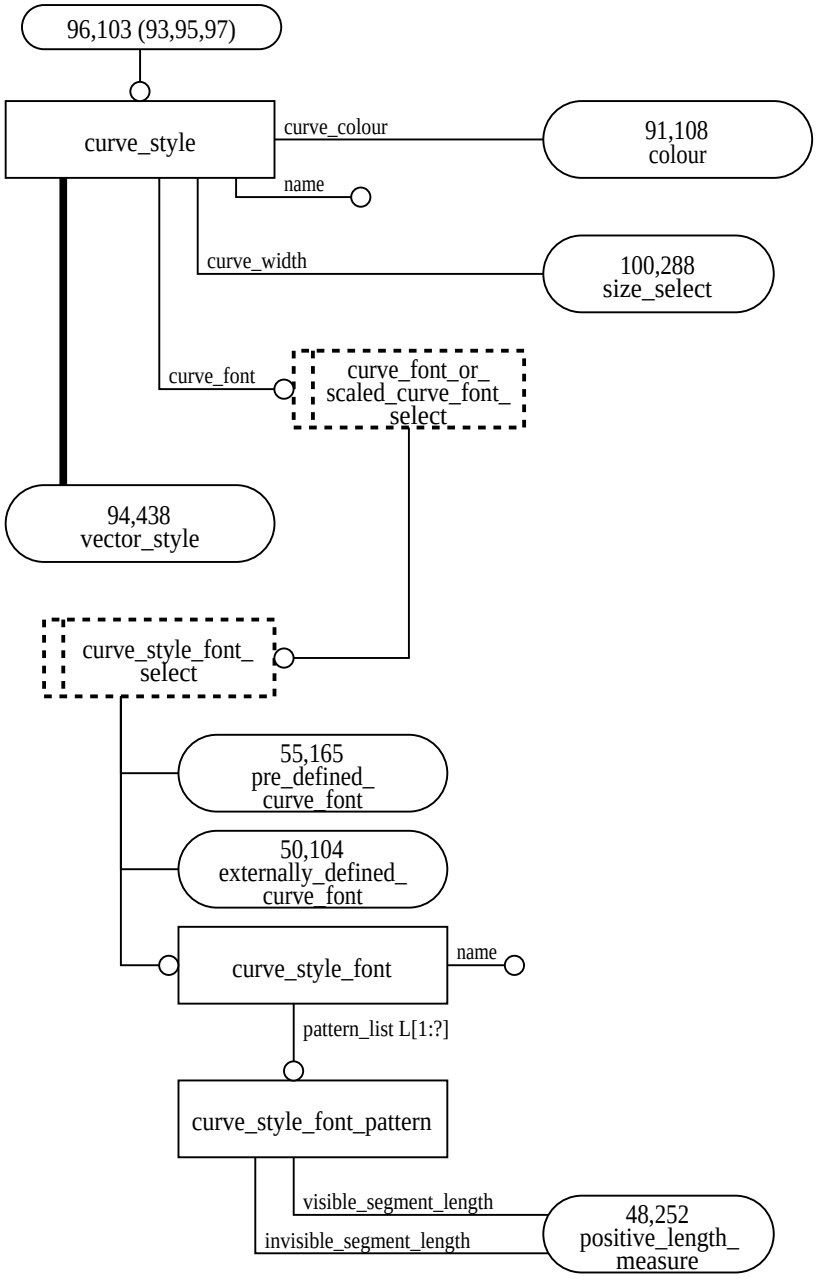


Figure H.96 – AIM diagram in EXPRESS_G: 96 of 124

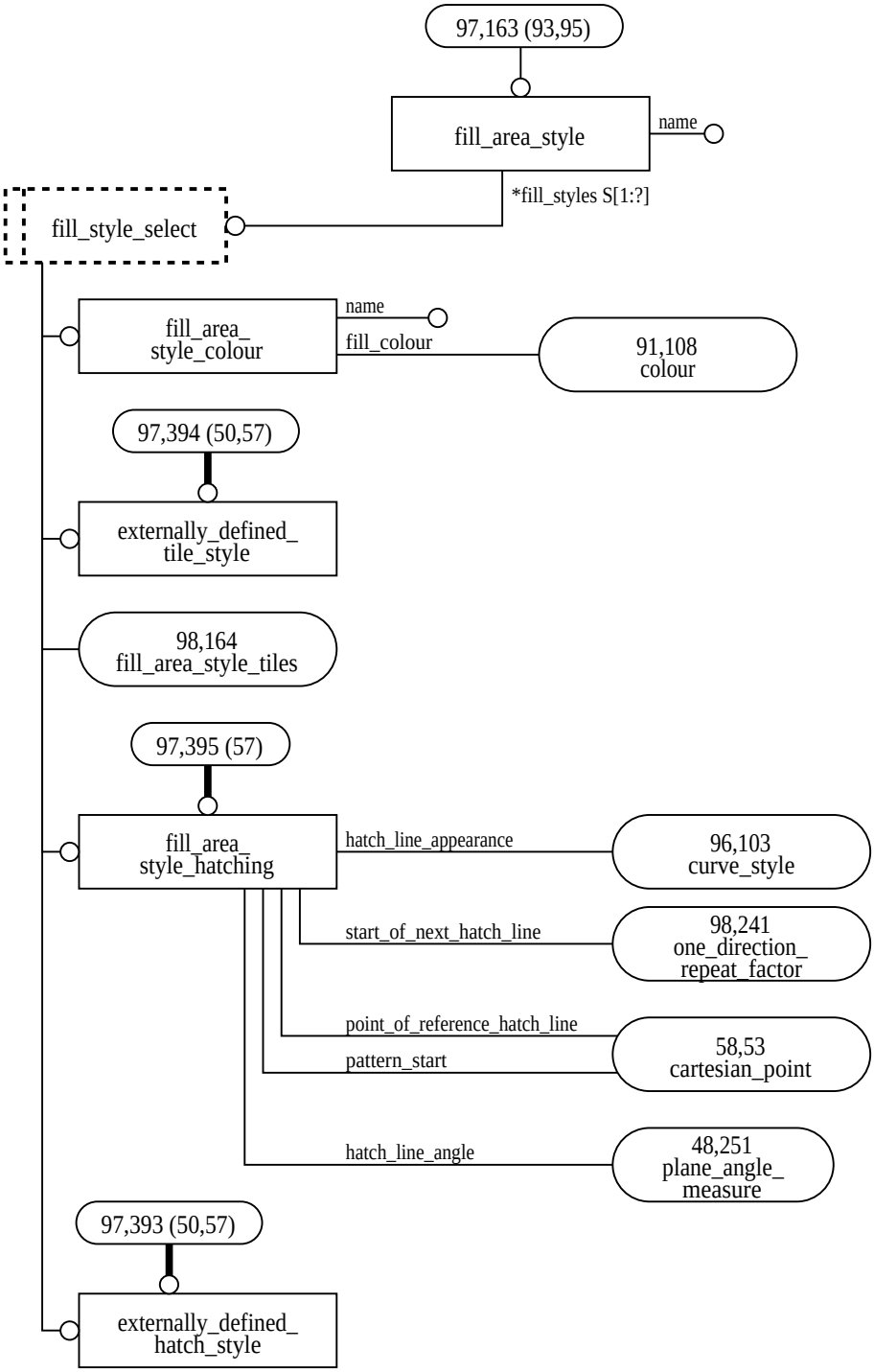


Figure H.97 – AIM diagram in EXPRESS_G: 97 of 124

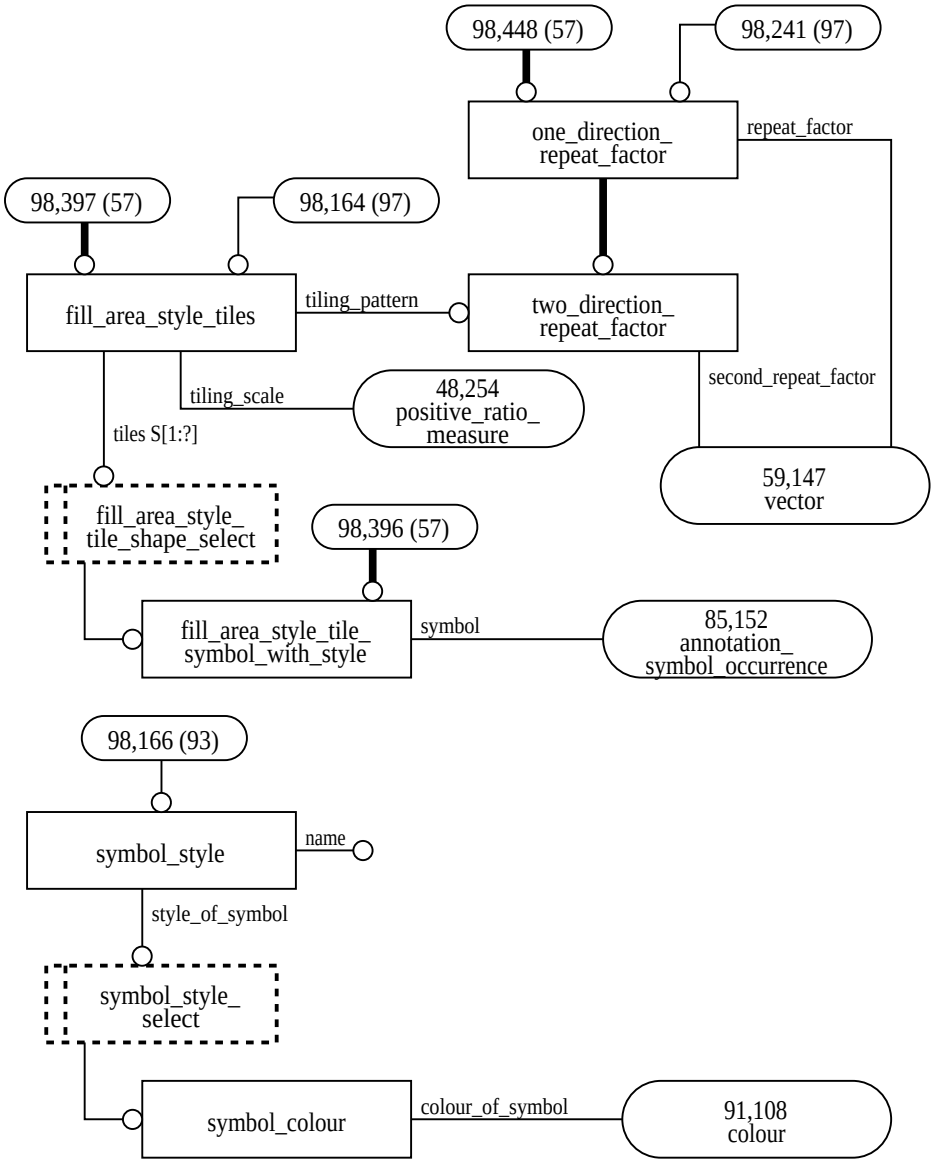


Figure H.98 – AIM diagram in EXPRESS_G: 98 of 124

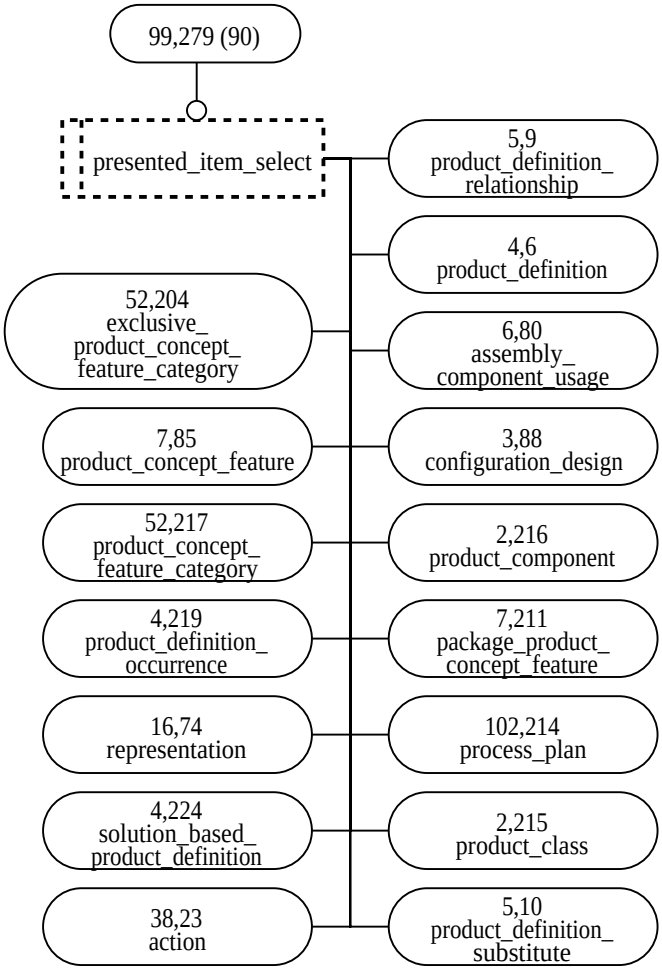


Figure H.99 – AIM diagram in EXPRESS_G: 99 of 124

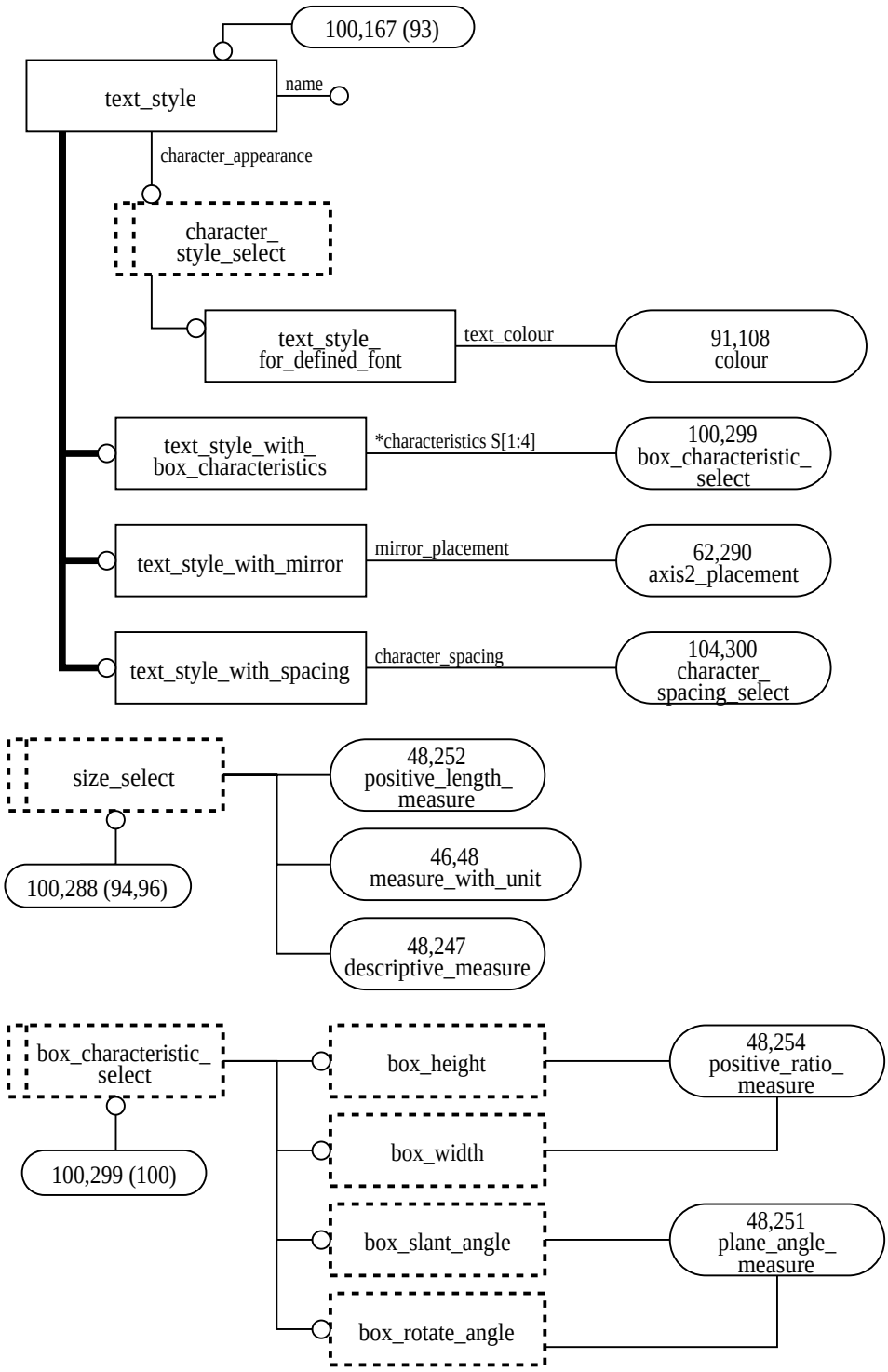


Figure H.100 – AIM diagram in EXPRESS_G: 100 of 124

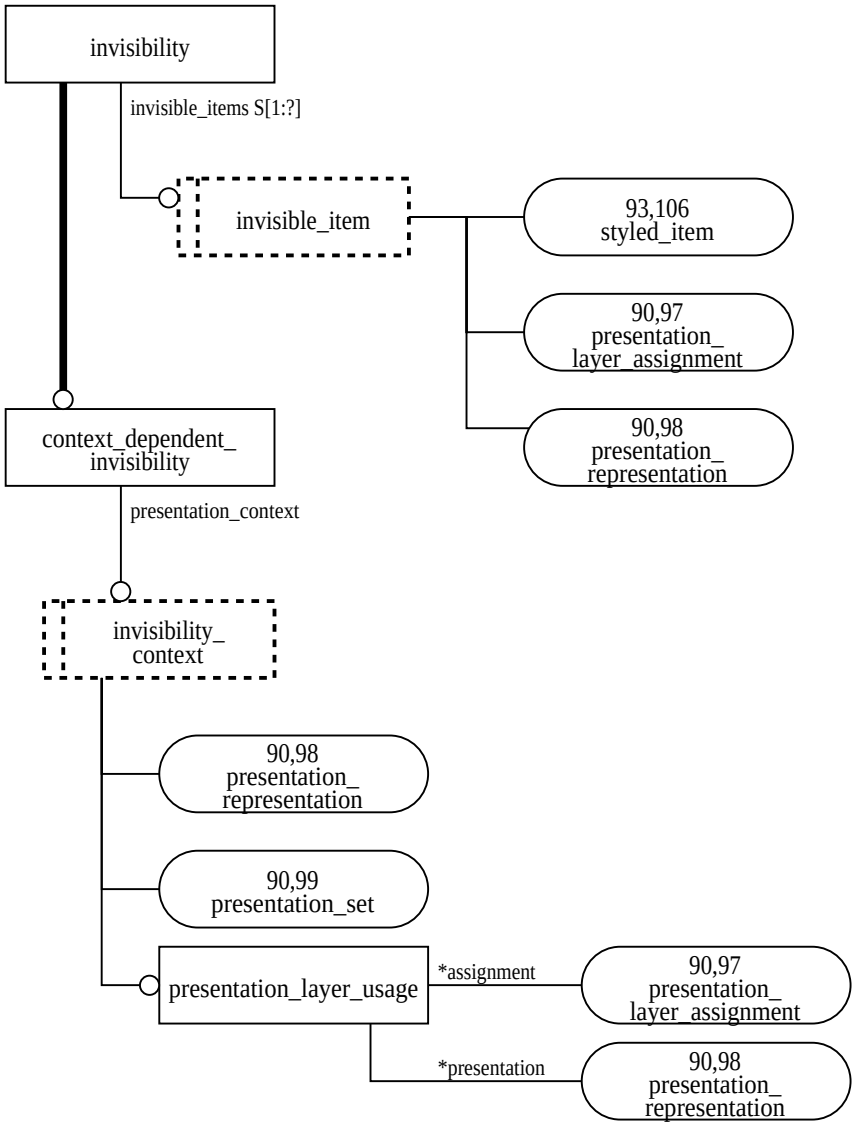


Figure H.101 – AIM diagram in EXPRESS_G: 101 of 124

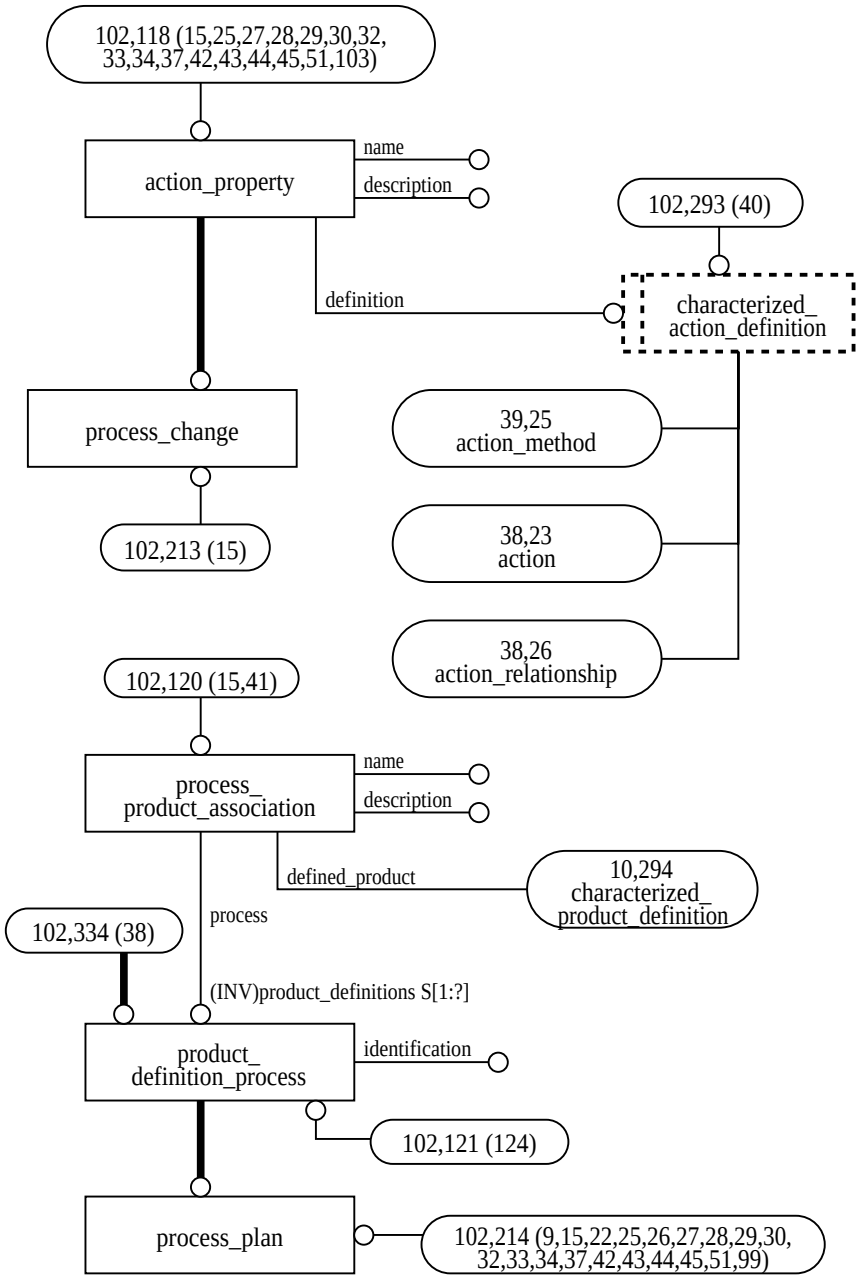


Figure H.102 – AIM diagram in EXPRESS_G: 102 of 124

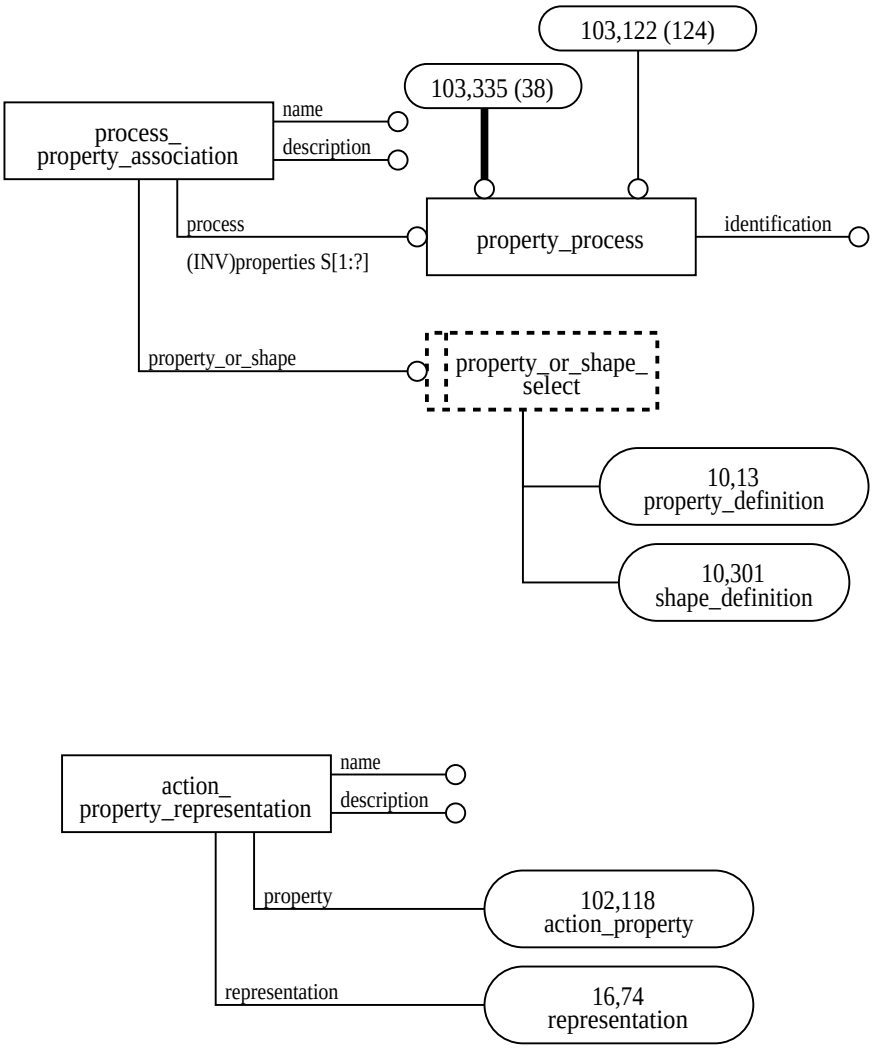


Figure H.103 – AIM diagram in EXPRESS_G: 103 of 124

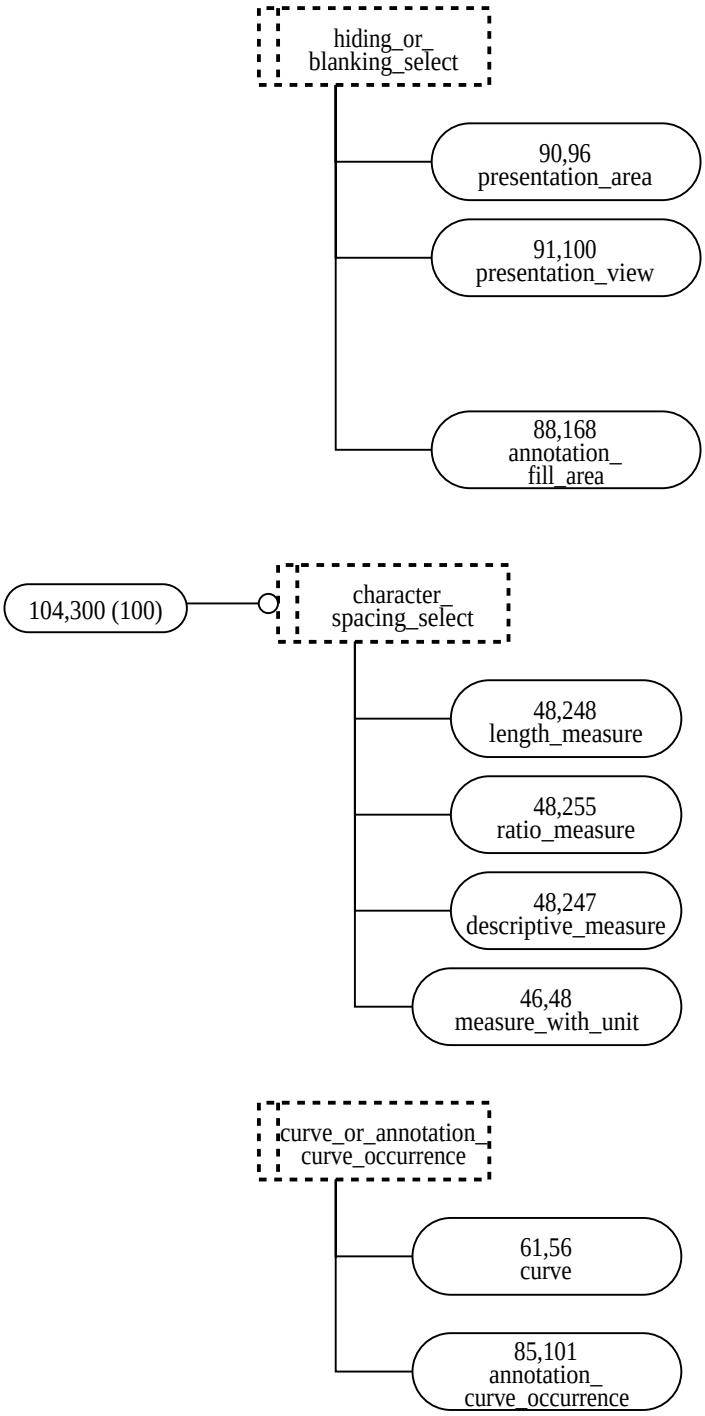


Figure H.104 – AIM diagram in EXPRESS_G: 104 of 124

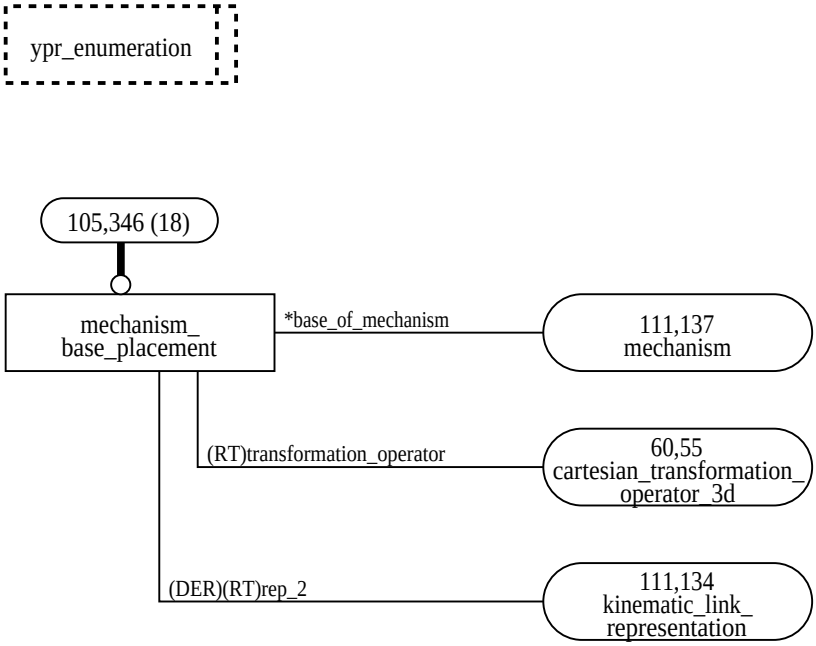


Figure H.105 – AIM diagram in EXPRESS_G: 105 of 124

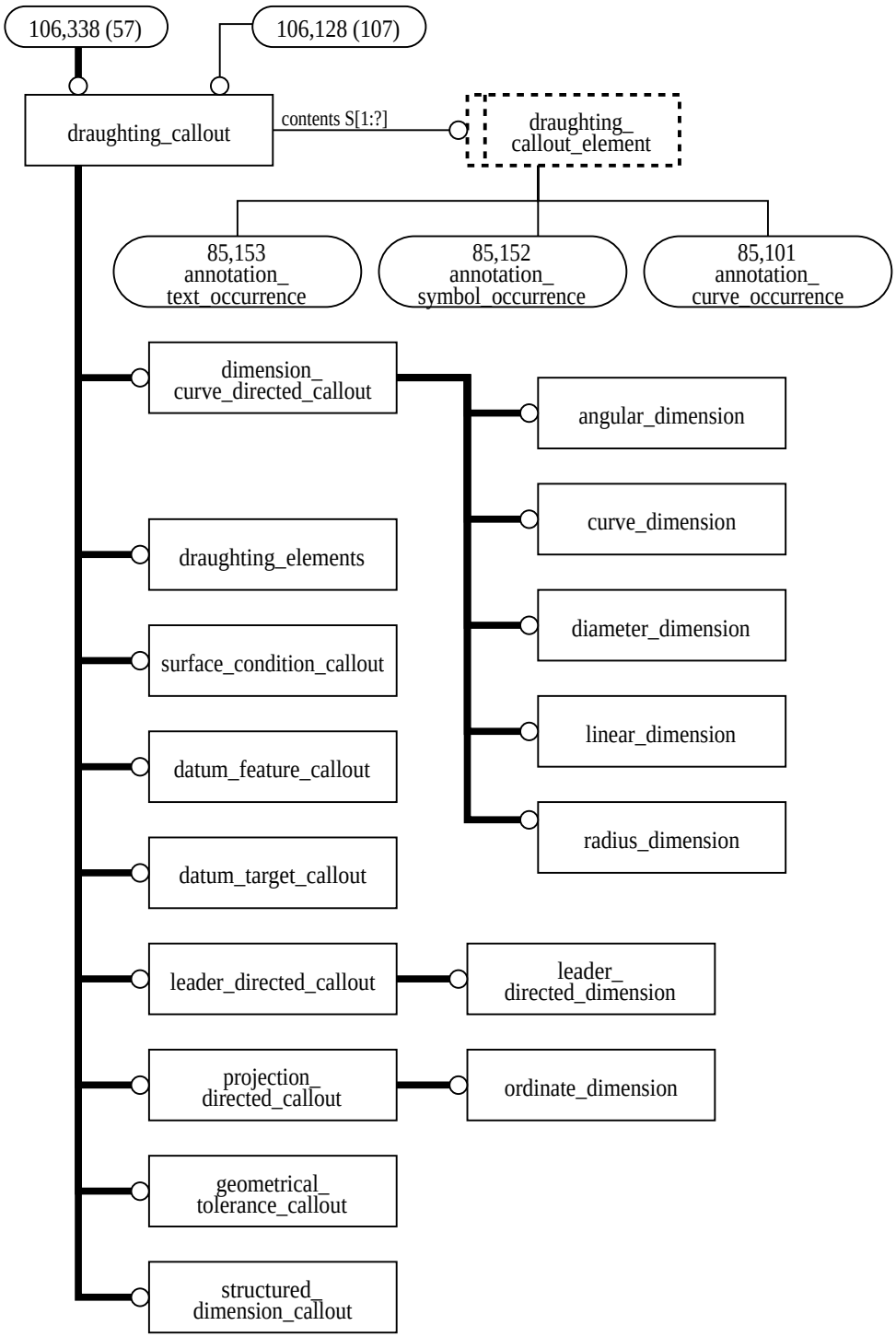


Figure H.106 – AIM diagram in EXPRESS_G: 106 of 124

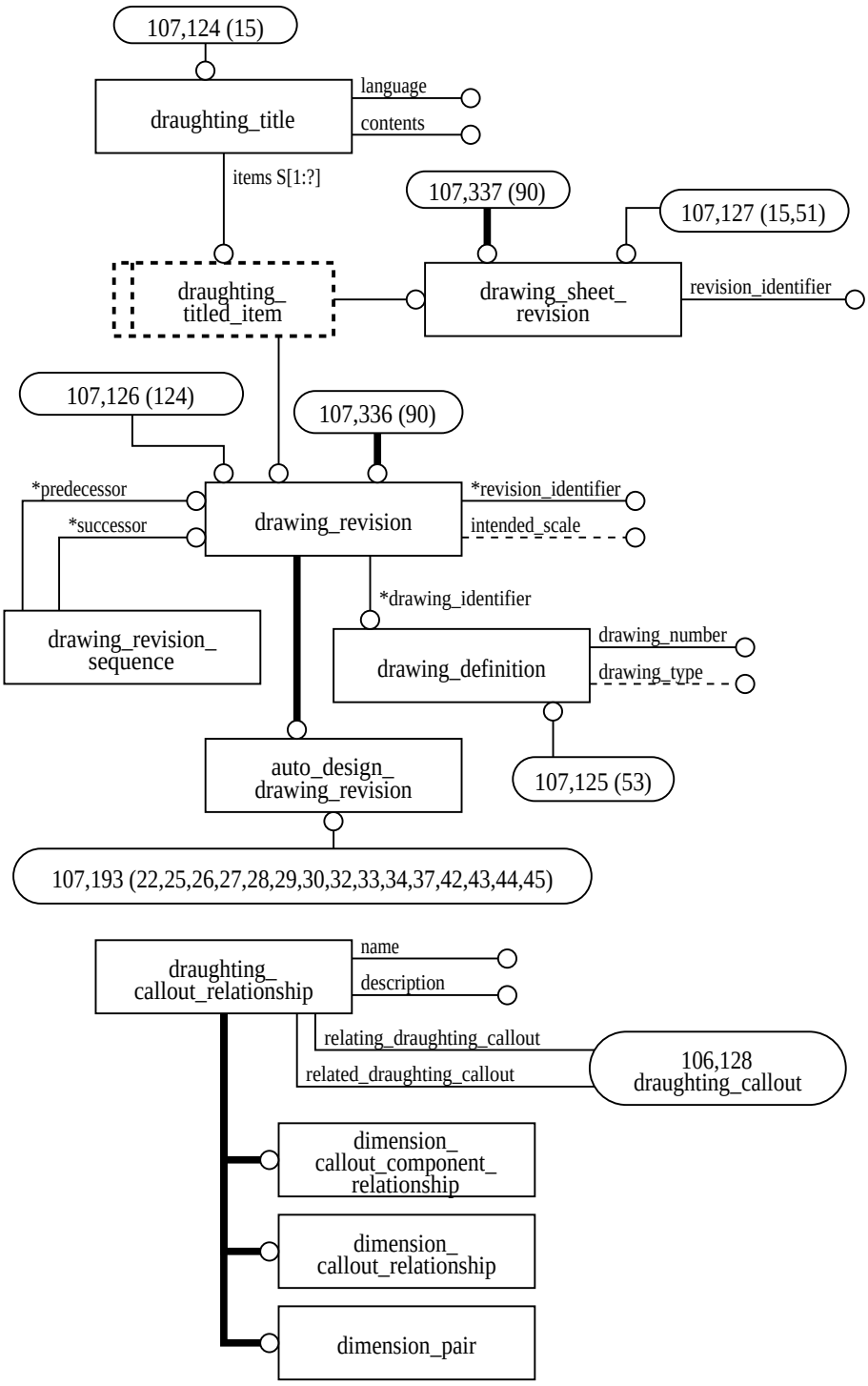


Figure H.107 – AIM diagram in EXPRESS_G: 107 of 124

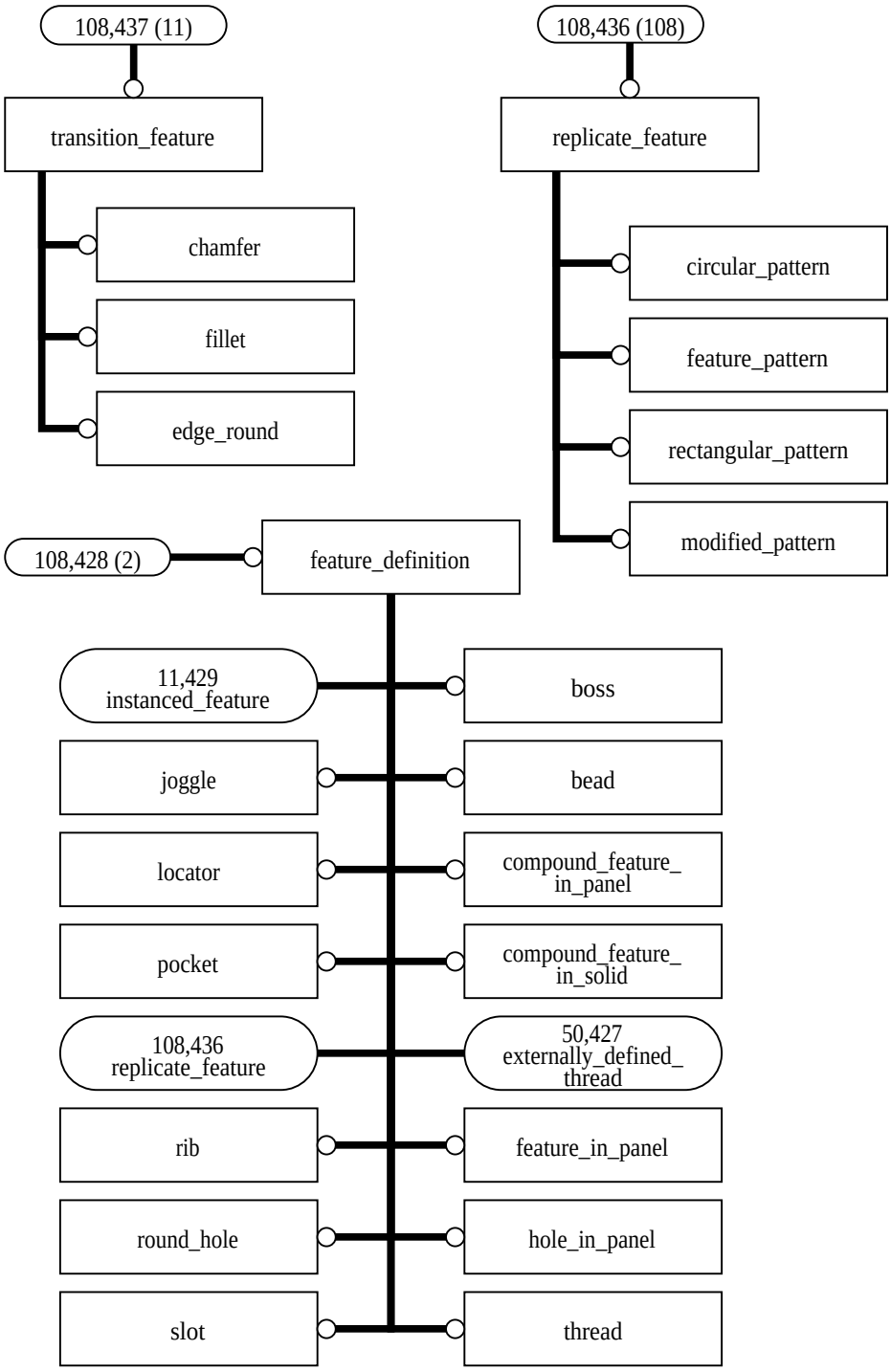


Figure H.108 – AIM diagram in EXPRESS_G: 108 of 124

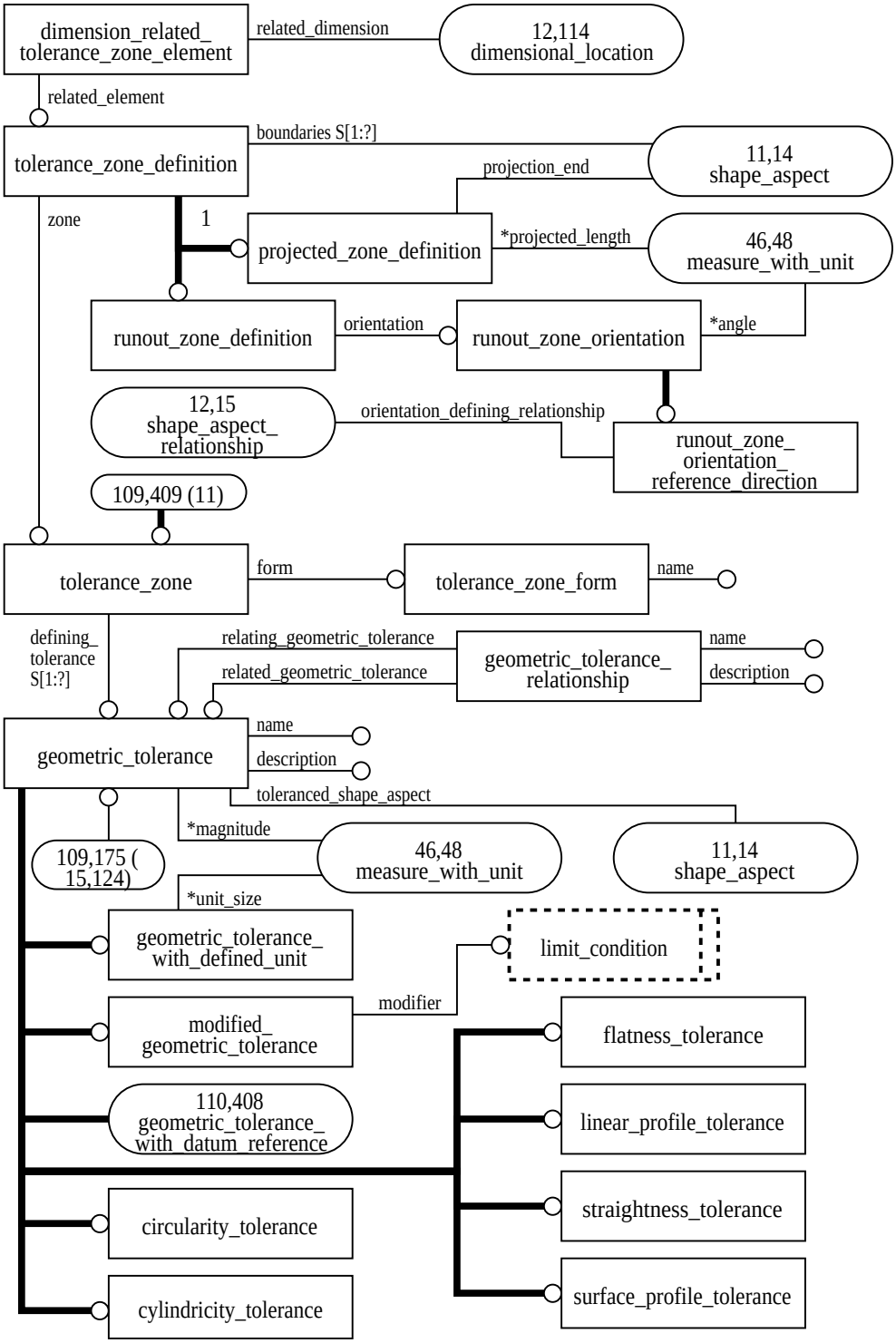


Figure H.109 – AIM diagram in EXPRESS_G: 109 of 124

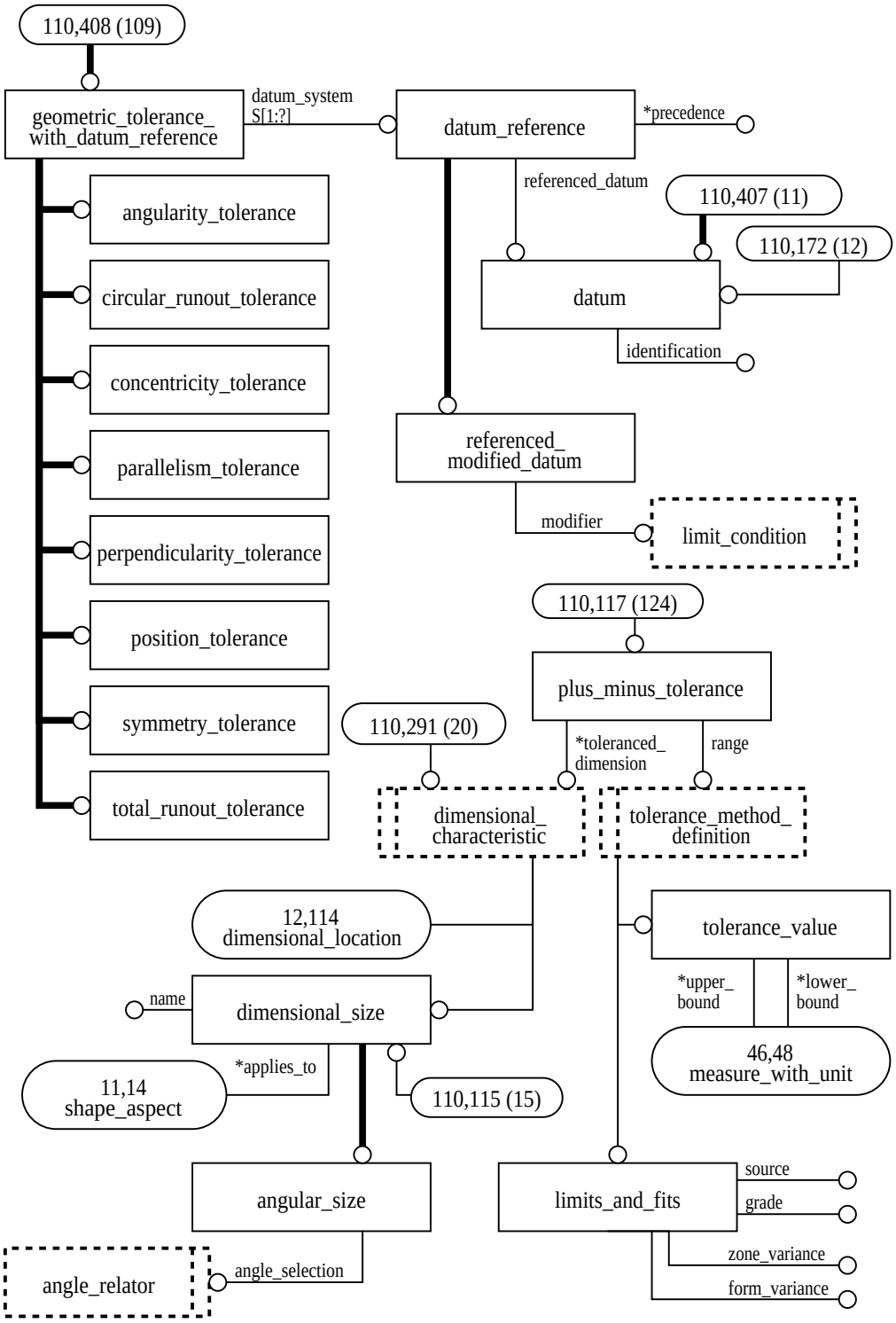


Figure H.110 – AIM diagram in EXPRESS_G: 110 of 124

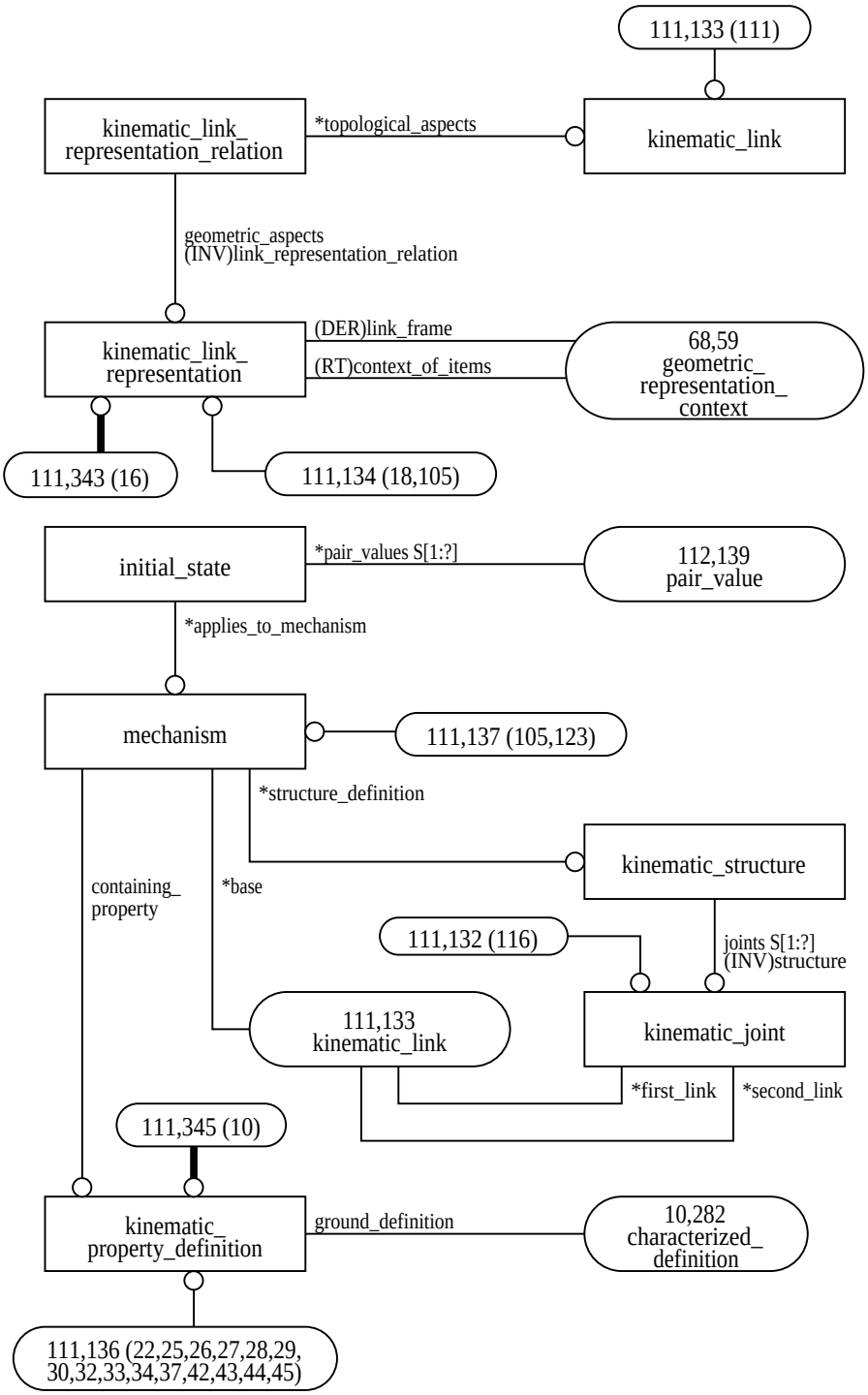


Figure H.111 – AIM diagram in EXPRESS_G: 111 of 124

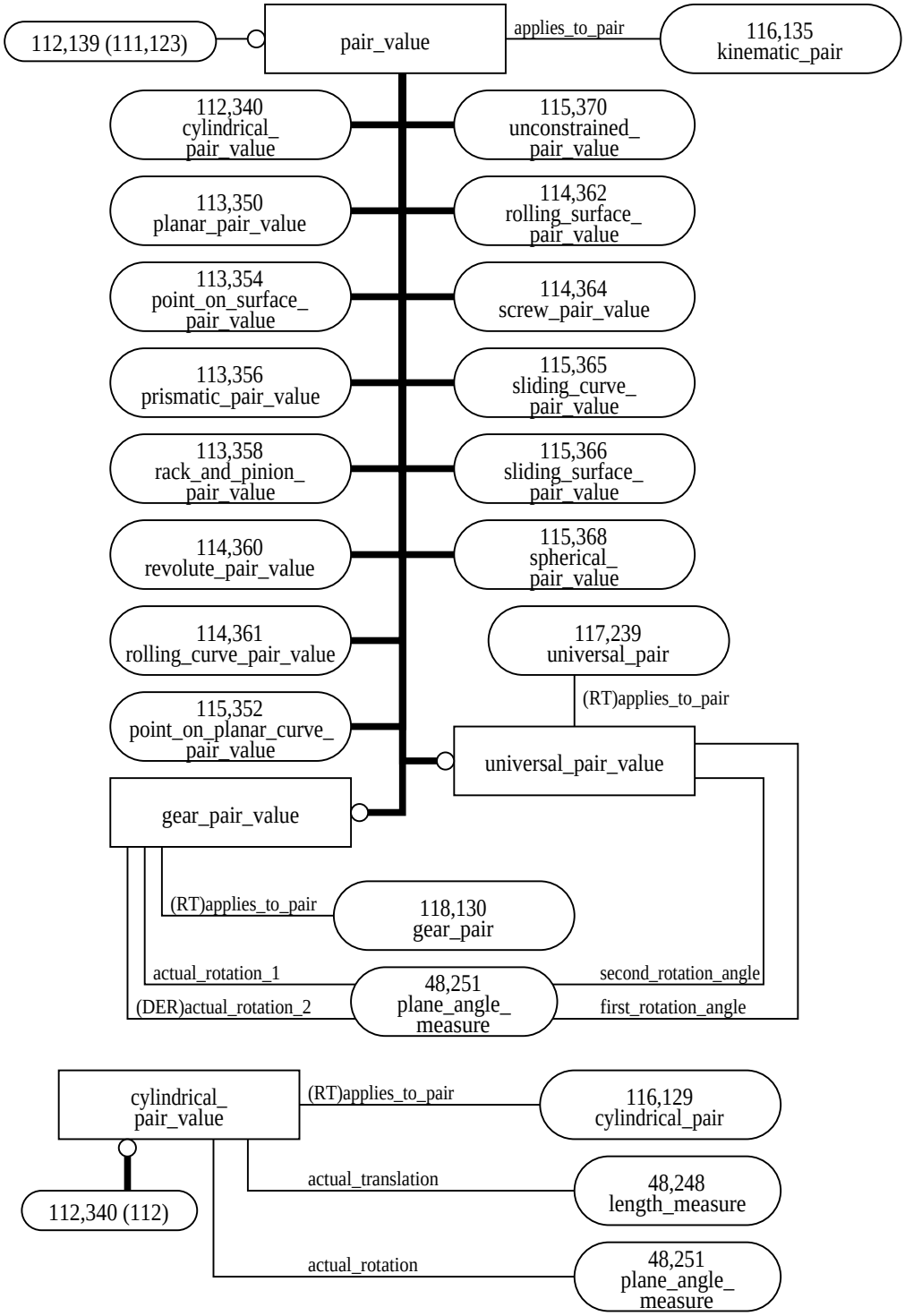


Figure H.112 – AIM diagram in EXPRESS_G: 112 of 124

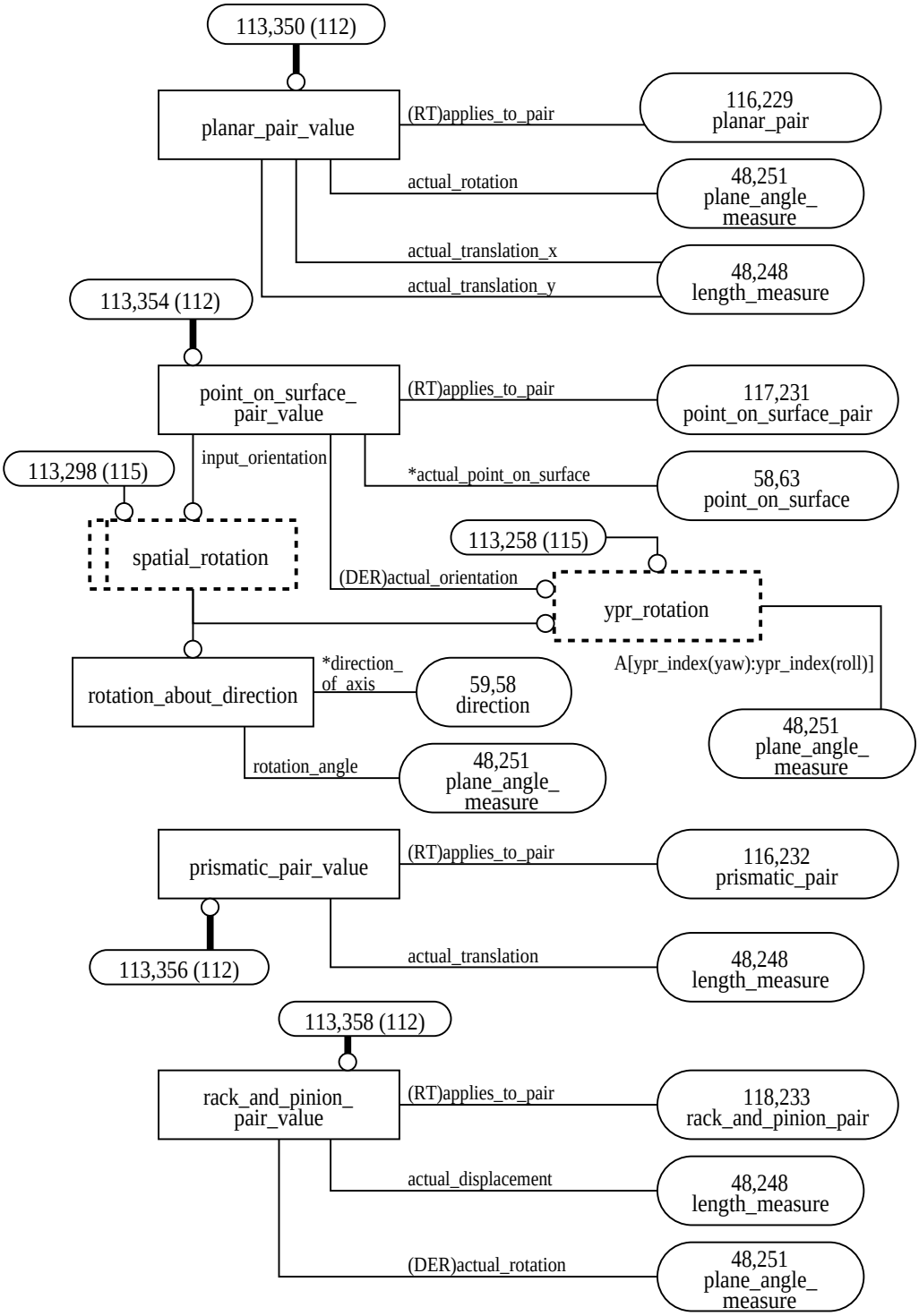


Figure H.113 – AIM diagram in EXPRESS_G: 113 of 124

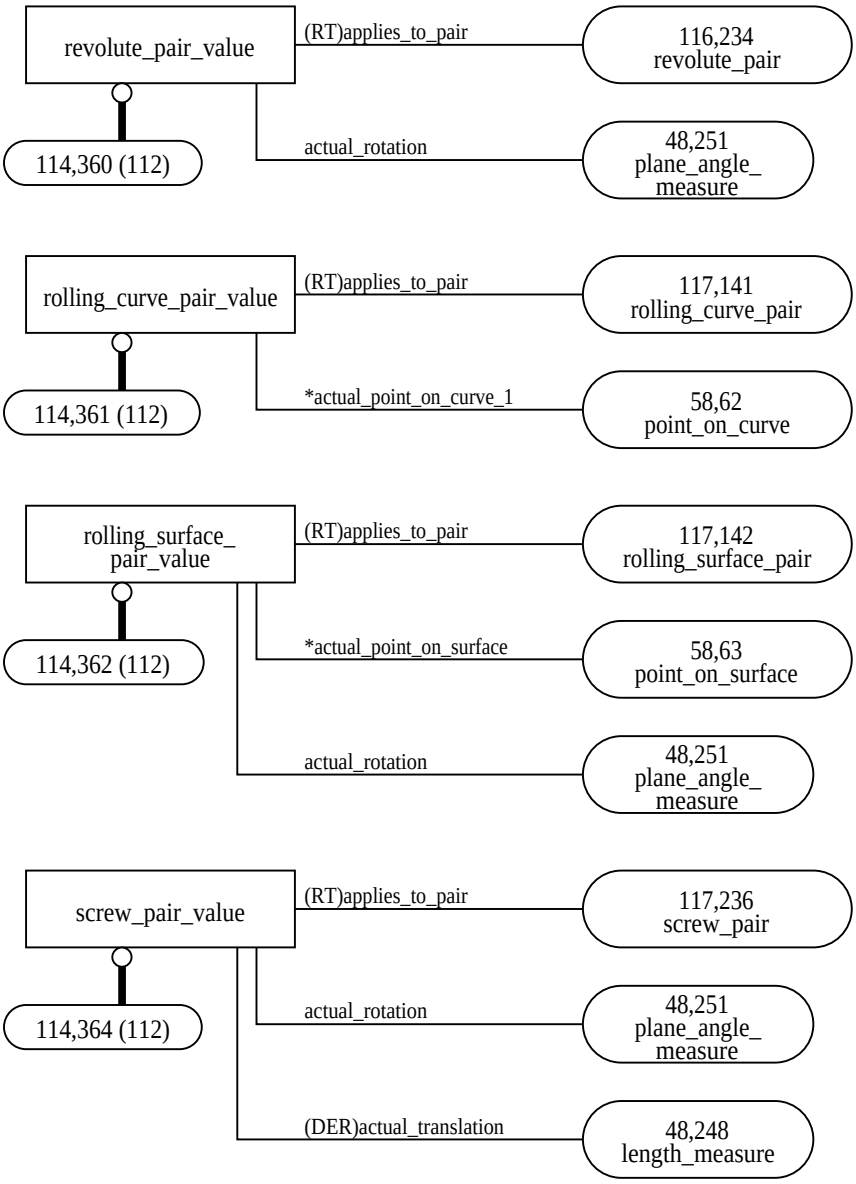


Figure H.114 – AIM diagram in EXPRESS_G: 114 of 124

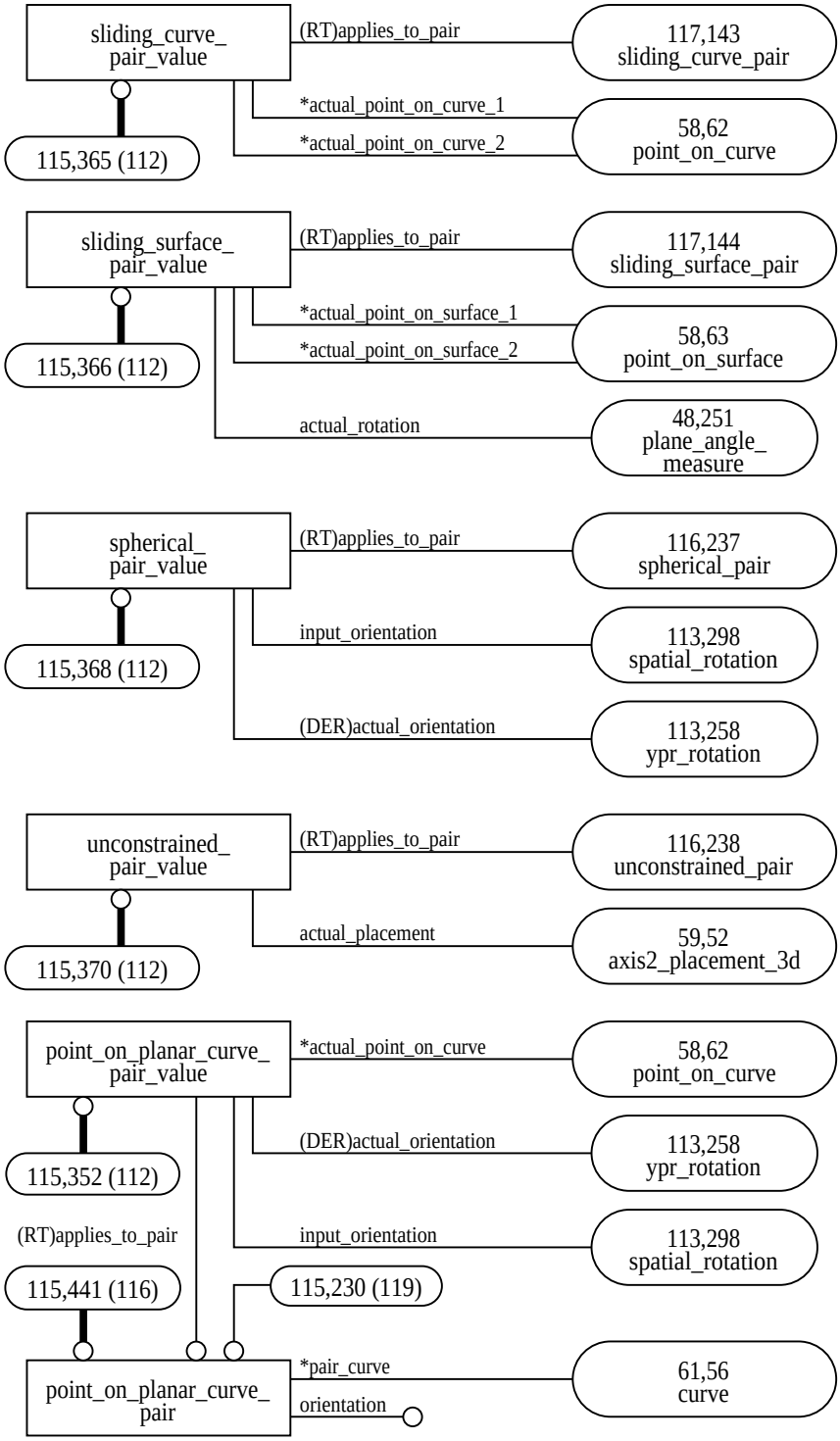


Figure H.115 – AIM diagram in EXPRESS_G: 115 of 124

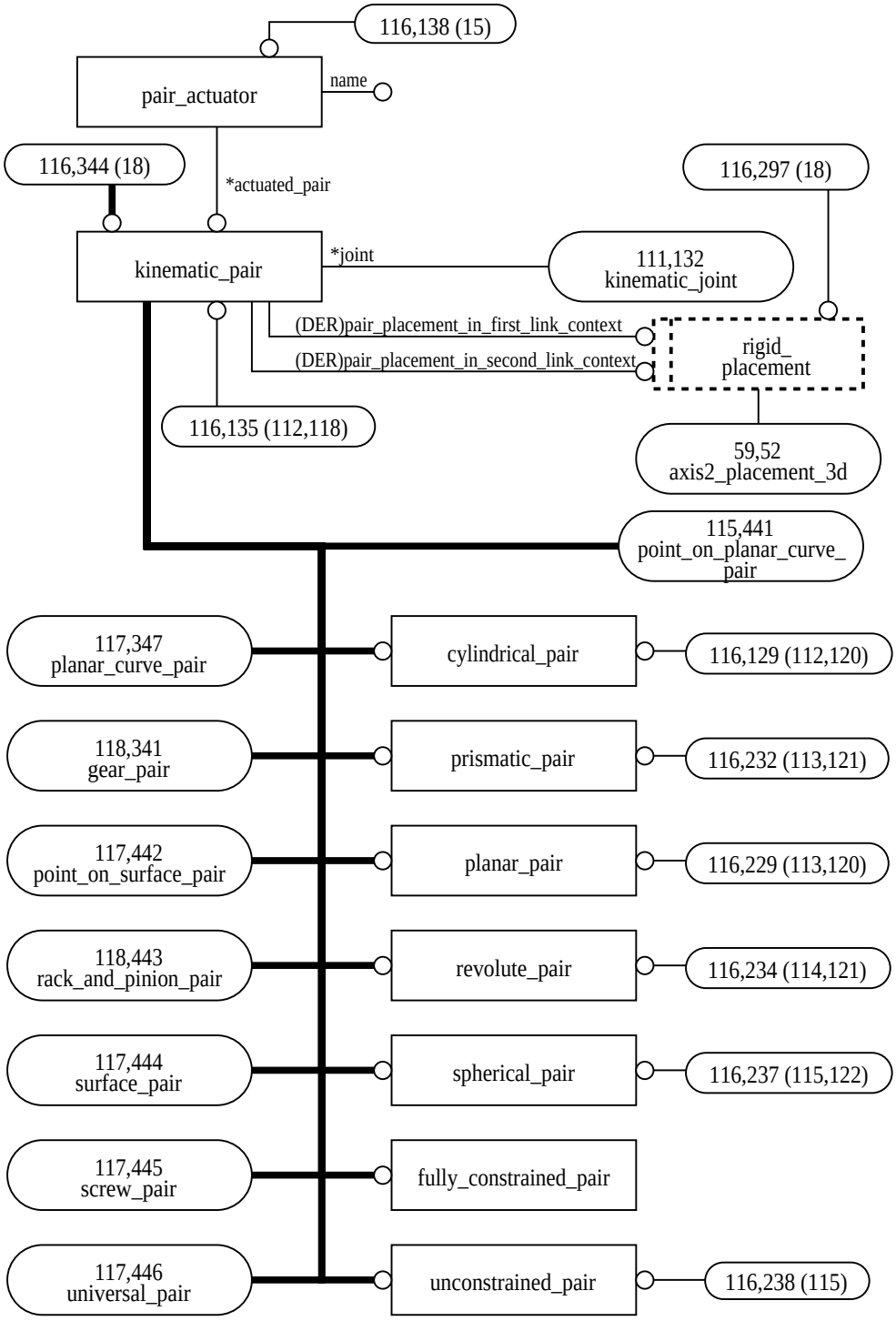


Figure H.116 – AIM diagram in EXPRESS_G: 116 of 124

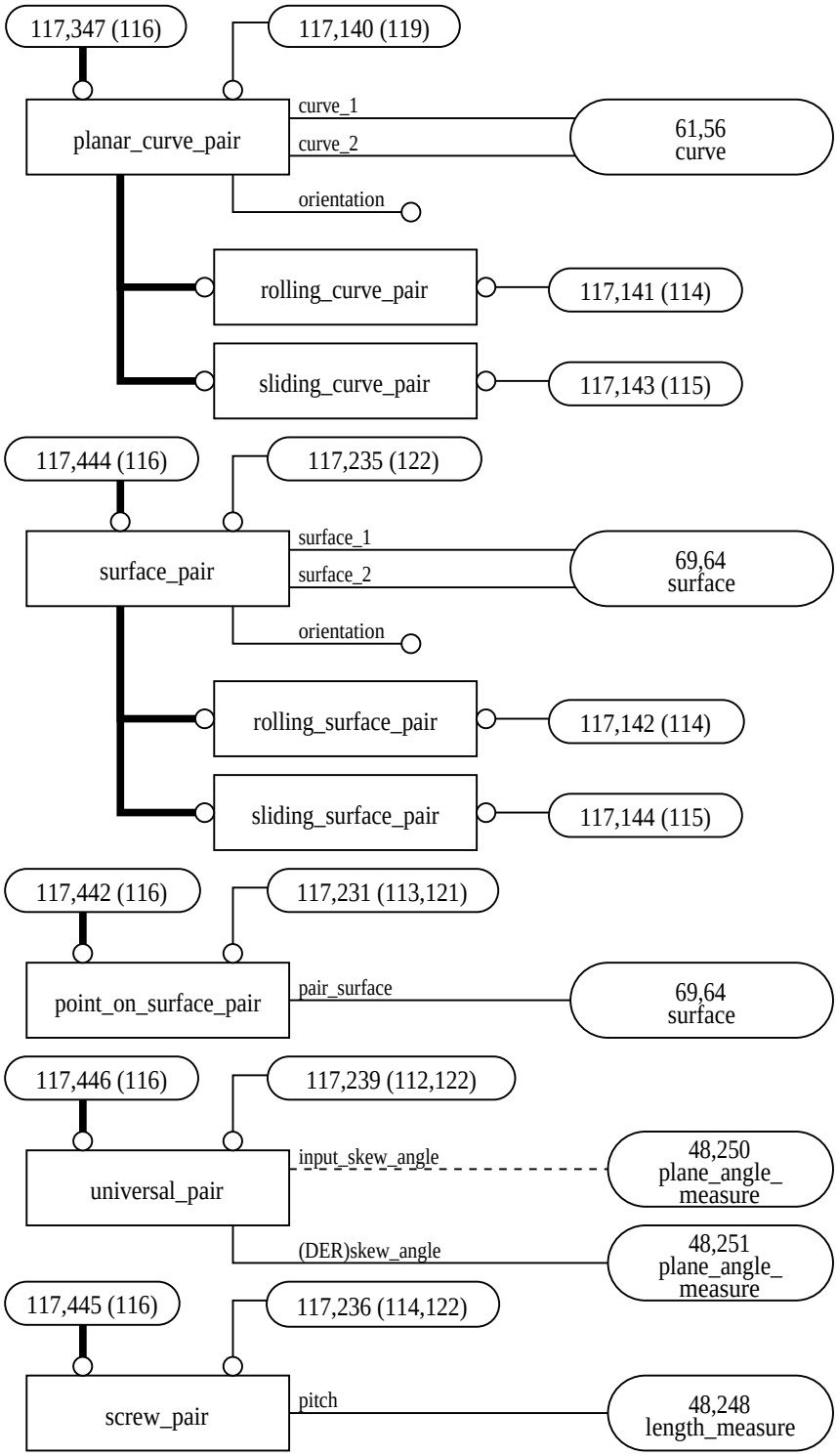


Figure H.117 – AIM diagram in EXPRESS_G: 117 of 124

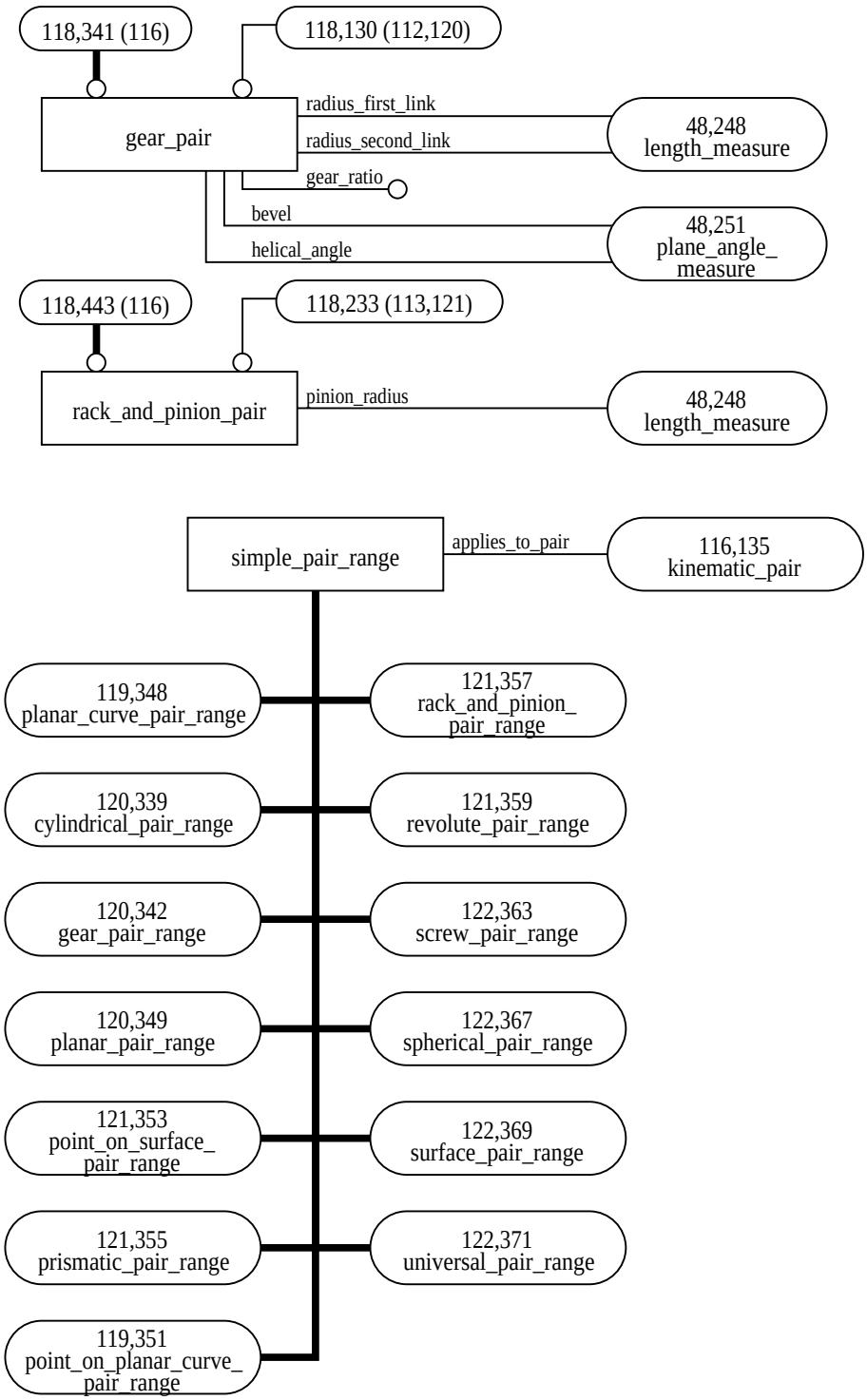


Figure H.118 – AIM diagram in EXPRESS_G: 118 of 124

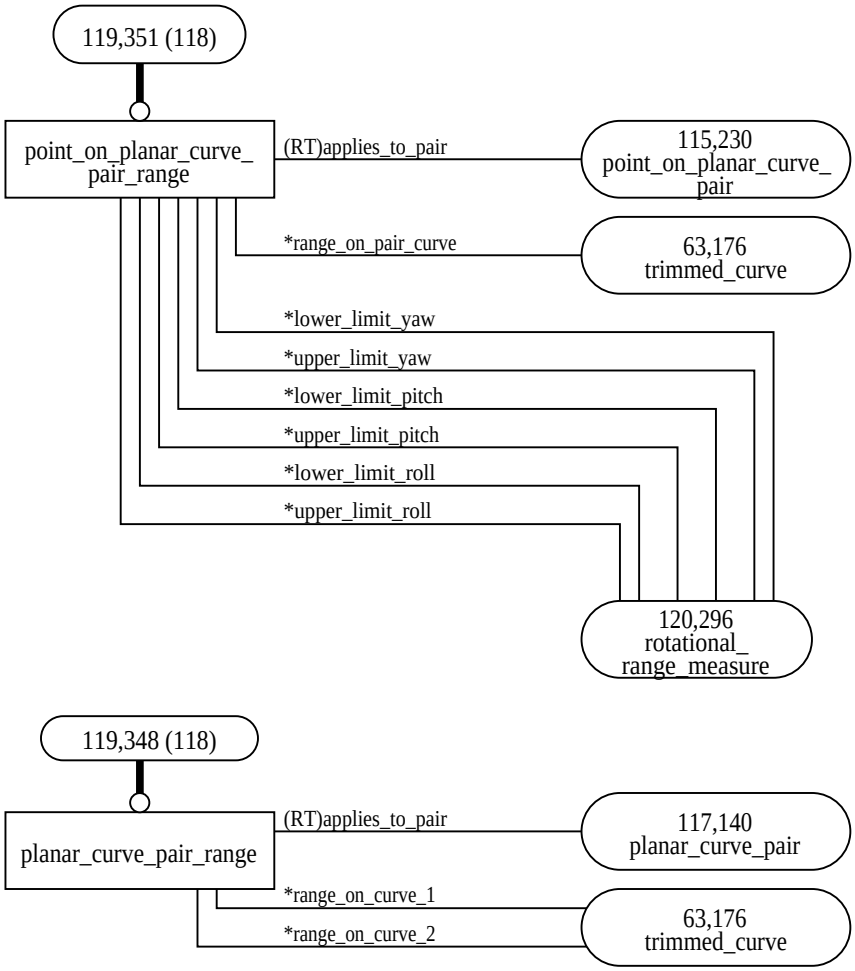


Figure H.119 – AIM diagram in EXPRESS_G: 119 of 124

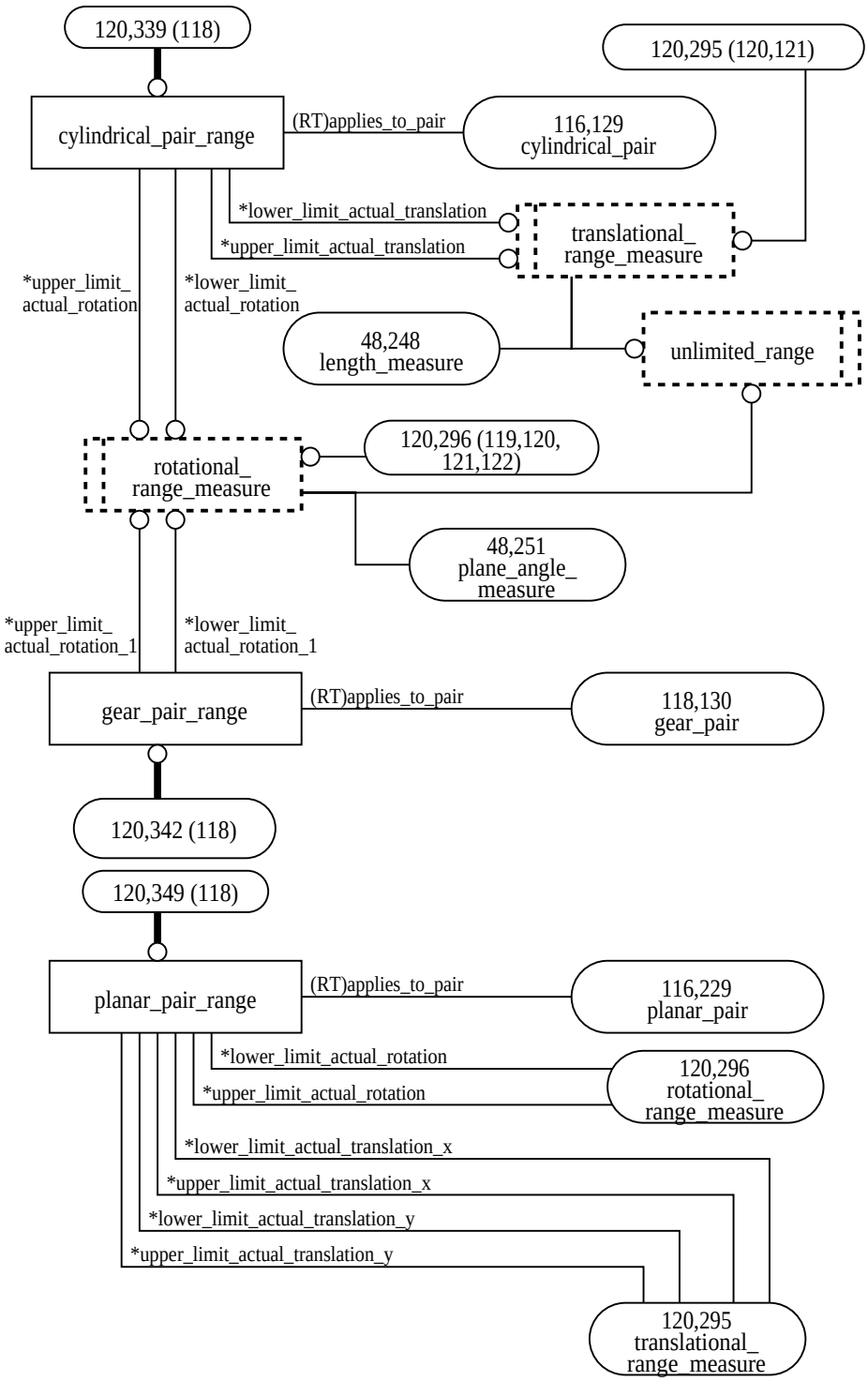


Figure H.120 – AIM diagram in EXPRESS_G: 120 of 124

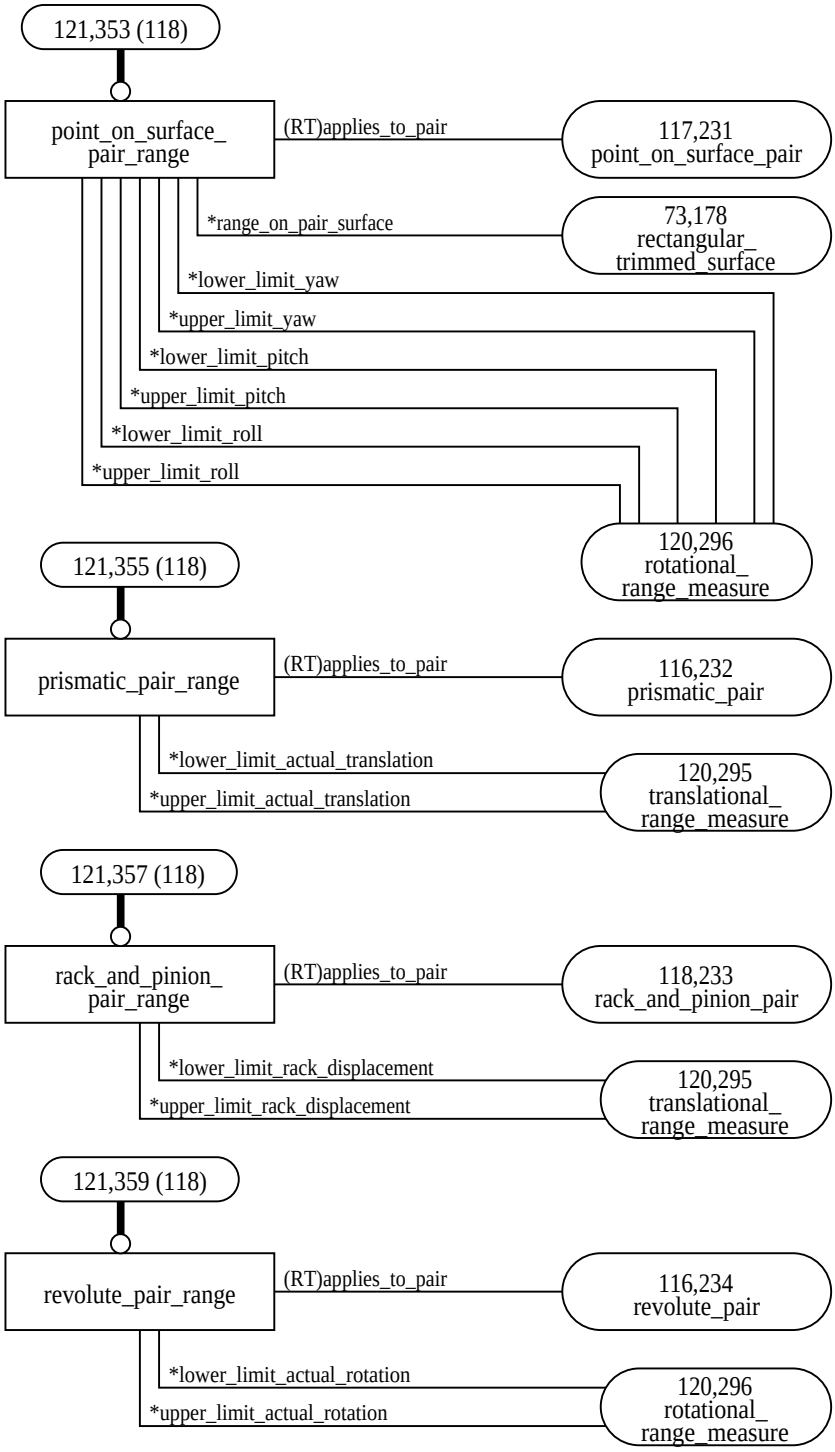


Figure H.121 – AIM diagram in EXPRESS_G: 121 of 124

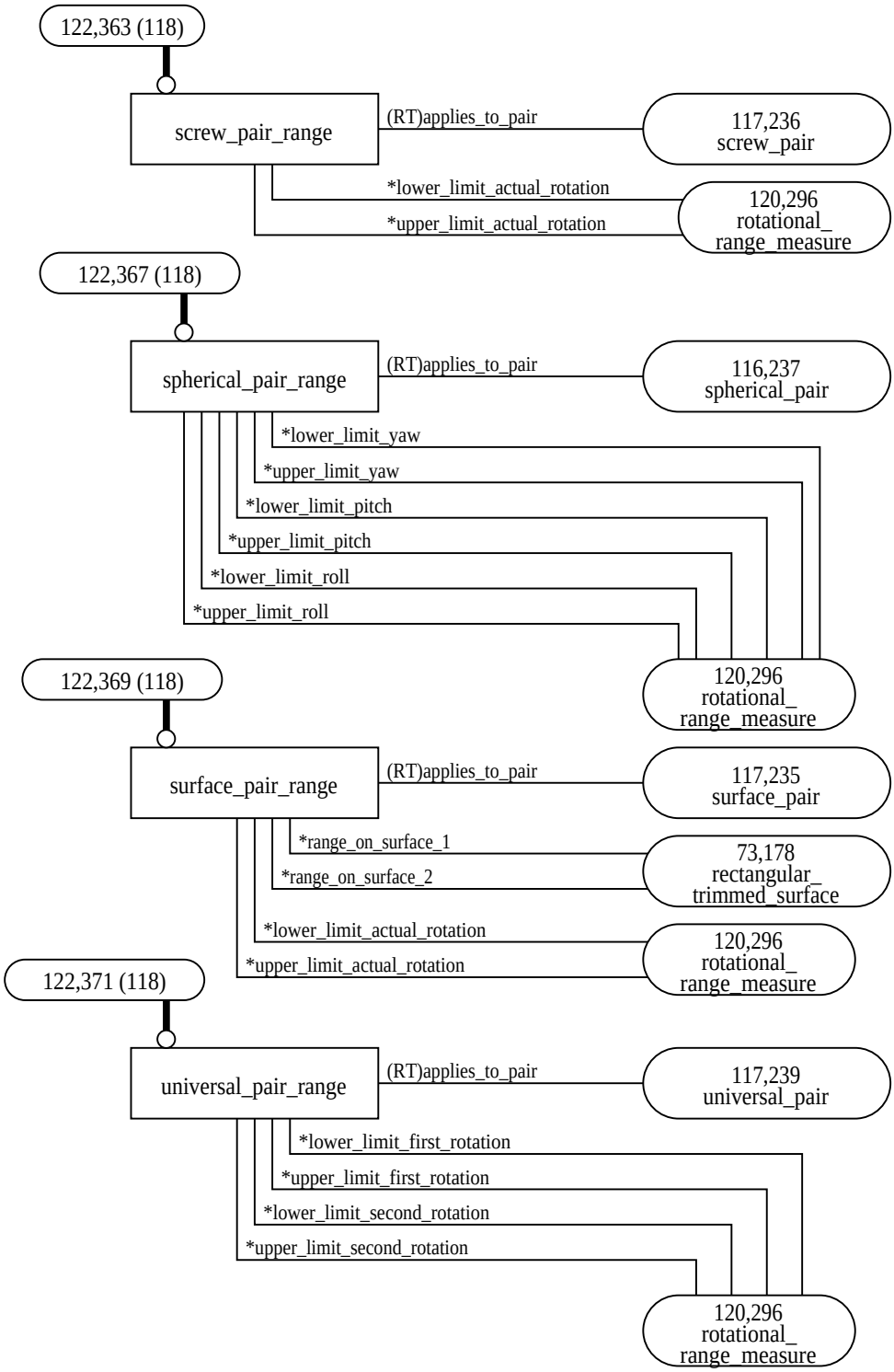


Figure H.122 – AIM diagram in EXPRESS_G: 122 of 124

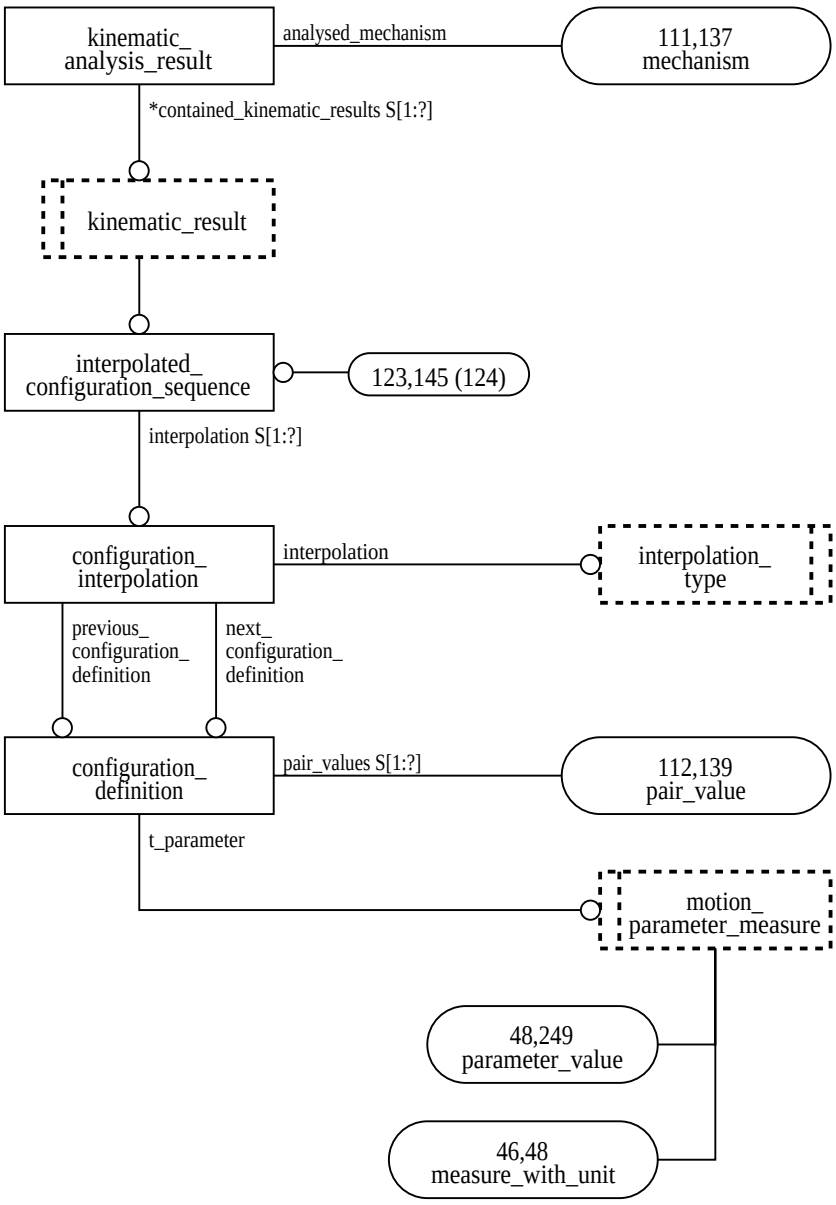


Figure H.123 – AIM diagram in EXPRESS_G: 123 of 124

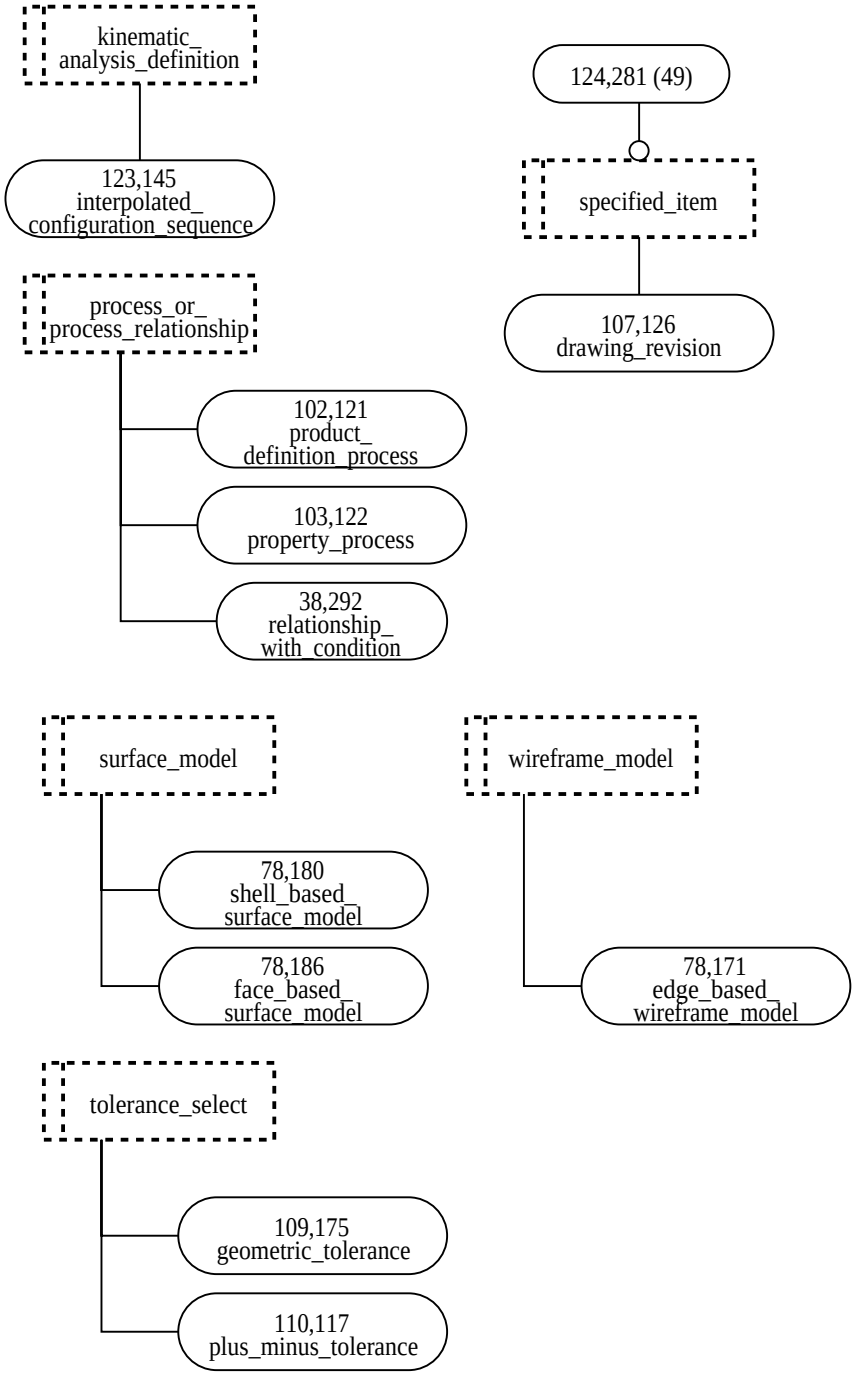


Figure H.124 – AIM diagram in EXPRESS_G: 124 of 124

Annex J
(informative)

AIM EXPRESS listing

This annex provides a listing of the table of short names and a listing of the EXPRESS specified in the AIM of this part of ISO 10303. No text or annotation is included. This annex is provided only in computer-interpretable form.

NOTE 4 – The information provided on this diskette is informative; the normative text is that contained in the body of this part of ISO 10303.

The expanded listing that is included in this annex corresponds to conformance class 18 as specified in clause 6 and to the **automotive_design** schema listed in annex A.

This annex also provides the listings of the EXPRESS of the conformance classes 1 (CC1) and 2 (CC2) as specified in clause 6 in computer-interpretable form. The schema names and object identifiers (see annex E) are as follows:

– CC1: Conformance class for component design with 3D shape representation

For the identification of the CC1 schema the object identifier

{ iso standard 10303 part(214) version(-2) object(1) automotive-design-cc01(3) }

is assigned to the **automotive_design_cc01** schema.

– CC2: Conformance class schema for assembly design with 3D shape representation

For the identification of the CC2 schema the object identifier

{ iso standard 10303 part(214) version(-2) object(1) automotive-design-cc02(4) }

is assigned to the **automotive_design_cc02** schema.

Table J.1 shows the correspondance between the file names on the diskette and the schemas they contain:

NOTE 5 – The single files are compressed into one file named 'a_d.zip'.

Table J.1 – File names for schemas

Information	File name
list of short names	s_names.txt
automotive_design schema	a_d.exp
automotive_design_cc01 schema	a_d_cc01.exp
automotive_design_cc02 schema	a_d_cc02.exp

Annex K
(informative)

ARM EXPRESS listing

The following EXPRESS-listing corresponds to the EXPRESS-G diagrams given in annex G. This annex is provided in computer-interpretable form only.

NOTE 1 – The entities are listed in alphabetical order. The type UNDEFINED_OBJECT refers to a user defined type, which is not detailed any further to be either of type INTEGER, REAL, STRING, etc. This level of detail was considered not to be appropriate the ARM modelling.

NOTE 2 – The information provided on this diskette is informative; the normative text is that contained in the body of this part of ISO 10303.

Annex L
(informative)

Bibliography

[1] *FIPS-93 Federal Information Processing Standard Publication 183 – Integration Definition for Function Modeling (IDEF0). FIPS PUB 183.* National Institute of Standards and Technology, USA, December 1993.

[2] *ISO/IEC 646:1991 Information processing – ISO 7-bit coded character set for information interchange.* International Organization for Standardization, 1991.

[3] *ISO 4287–1:1984 Surface roughness - Terminology - Part1: Surface and its parameters.* International Organization for Standardization, 1984

[4] *ISO 4516:1980 Metallic and related coatings; Vickers and Knoop microhardness tests.* International Organization for Standardization, 1980

[5] *ISO 6508:1986 Metallic materials – Hardness test – Rockwell test (scales A - B - C - D - E - F - G - H - K).* International Organization for Standardization, 1986

[6] *ISO 6983:1982 Numerical control of machines – program format and definitions of address words* International Organization for Standardization, 1982

[7] *ISO 8859–1:1987 Information processing; 8-bit-single-byte coded graphic character sets; part 1: Latin alphabet No. 1.* International Organization for Standardization, 1987

[8] *ISO 8879:1986 Information processing – Text and office systems – Standard Generalized Markup Language (SGML).* International Organization for Standardization, 1986

[9] ODETTE *ODG 11: CAD/CAM quality assurance method.* ODETTE.

[10] VDA *VDA4955: CAD/CAM data quality.* Arbeitskreis 'CAD/CAM' im VDA Rohstoff-Ausschuss (VDA-AK 'CAD/CAM').

Index

Accuracy
- application assertion ... 397
- application object ... 58
- ARM EXPRESS-G (figure G.41) ... 2408
- mapping table ... 1161

action
- AIM expanded listing entities ... 2002
- AIM EXPRESS-G entity (figure H.38) ... 2457
- AIM short listing imported entity modifications ... 1735
- mapping table ... 1227, 1244, 1267, 1360, 1384, 1394, 1410, 1411, 1419

action_assignment
- AIM expanded listing entities ... 2002
- AIM EXPRESS-G entity (figure H.41) ... 2460
- AIM short listing imported entity modifications ... 1736

action_assignment_subtype_exclusiveness
- AIM expanded listing rules ... 2254
- AIM short listing rules ... 1766
- mapping table ... 1312

action_directive
- AIM expanded listing entities ... 2002
- AIM EXPRESS-G entity (figure H.39) ... 2458
- AIM short listing imported entity modifications ... 1736
- mapping table ... 1309, 1310

action_item
- AIM expanded listing types ... 1980
- AIM EXPRESS-G types (figure H.43) ... 2462
- AIM short listing types ... 1497

action_method
- AIM expanded listing entities ... 2002
- AIM EXPRESS-G entity (figure H.39) ... 2458
- AIM short listing imported entity modifications ... 1736
- mapping table ... 1288, 1360, 1394

action_property
- AIM expanded listing entities ... 2002
- AIM EXPRESS-G entity (figure H.102) ... 2521
- AIM short listing imported entity modifications ... 1736
- mapping table ... 1039, 1169, 1176, 1384, 1394, 1414

action_property_representation
- AIM expanded listing entities ... 2002
- AIM EXPRESS-G entity (figure H.103) ... 2522

action_relationship
- AIM expanded listing entities ... 2002
- AIM EXPRESS-G entity (figure H.38) ... 2457
- AIM short listing imported entity modifications ... 1737
- mapping table ... 1288, 1290, 1292, 1298, 1308, 1405, 1412

action_request_assignment
- AIM expanded listing entities ... 2003
- AIM EXPRESS-G entity (figure H.39) ... 2458

action_request_item
- AIM expanded listing types ... 1980

AIM EXPRESS-G types (figure H.42) 2461
AIM short listing types 1498
action_request_role
AIM expanded listing entities 2003
AIM EXPRESS-G entity (figure H.39) 2458
AIM short listing imported entity modifications 1737
action_request_solution
AIM expanded listing entities 2003
AIM EXPRESS-G entity (figure H.39) 2458
action_request_status
AIM expanded listing entities 2003
AIM EXPRESS-G entity (figure H.39) 2458
mapping table 1311
action_resource
AIM expanded listing entities 2003
AIM EXPRESS-G entity (figure H.40) 2459
action_resource_requirement
AIM expanded listing entities 2003
AIM EXPRESS-G entity (figure H.40) 2459
AIM short listing imported entity modifications 1737
mapping table 1407
action_resource_type
AIM expanded listing entities 2003
AIM EXPRESS-G entity (figure H.40) 2459
action_role
AIM expanded listing entities 2003
AIM EXPRESS-G entity (figure H.41) 2460
AIM short listing imported entity modifications 1737
mapping table 1274
action_status
AIM expanded listing entities 2004
AIM EXPRESS-G entity (figure H.38) 2457
mapping table 1267
action_subtype_restriction
AIM expanded listing rules 2254
AIM short listing rules 1767
Activity
application assertion . 397–399, 405, 410, 440, 442, 449, 475, 481, 530, 549, 554, 564, 573, 592
application object 59
ARM EXPRESS-G (figure G.17) 2384
mapping table 1267
Activity_element
application assertion 399–404, 410, 442, 449, 474, 475, 530, 554, 564
application object 63
ARM EXPRESS-G (figure G.16) 2383
mapping table 1274
Activity_method
application assertion 397, 399, 404, 592
application object 64
ARM EXPRESS-G (figure G.16) 2383
mapping table 1288
Activity_relationship

application assertion 405
application object 64
ARM EXPRESS-G (figure G.17) 2384
mapping table 1288
activity_requires_person_or_organization_assignment
AIM expanded listing rules 2254
AIM short listing rules 1768
mapping table 1127
Actuator
application assertion 405
application object 66
ARM EXPRESS-G (figure G.22) 2389
mapping table 970
acyclic_composite_text
AIM expanded listing functions 2204
acyclic_curve_replica
AIM expanded listing functions 2205
acyclic_mapped_item_usage
AIM expanded listing functions 2205
acyclic_mapped_representation
AIM expanded listing functions 2206
acyclic_point_replica
AIM expanded listing functions 2207
acyclic_product_category_relationship
AIM expanded listing functions 2207
acyclic_product_definition_relationship
AIM expanded listing functions 2207
acyclic_solid_replica
AIM expanded listing functions 2208
acyclic_surface_replica
AIM expanded listing functions 2208
address
AIM expanded listing entities 2004
AIM EXPRESS-G entity (figure H.24) 2443
advanced_brep_shape_representation
AIM expanded listing entities 2004
AIM EXPRESS-G entity (figure H.20) 2439
mapping table 898, 934
advanced_face
AIM expanded listing entities 2005
AIM EXPRESS-G entity (figure H.82) 2501
mapping table 873, 880, 899, 906, 916
ahead_or_behind
AIM expanded listing types 1981
AIM EXPRESS-G types (figure H.31) 2450
Alias_identification
application assertion 405–408
application object 66
ARM EXPRESS-G (figure G.4) 2371
mapping table 1048
Alternate_item_relationship
application assertion 408, 410

application object 67
ARM EXPRESS-G (figure G.5) 2372
mapping table 1289
alternate_product_relationship
AIM expanded listing entities 2006
AIM EXPRESS-G entity (figure H.6) 2425
AIM short listing imported entity modifications 1738
mapping table 1289
amount_of_substance_measure
AIM expanded listing types 1981
AIM EXPRESS-G types (figure H.48) 2467
amount_of_substance_measure_with_unit
AIM expanded listing entities 2006
AIM EXPRESS-G entity (figure H.46) 2465
amount_of_substance_unit
AIM expanded listing entities 2006
AIM EXPRESS-G entity (figure H.47) 2466
angle_relator
AIM expanded listing types 1981
AIM EXPRESS-G types (figure H.12) 2431
Angle_specification
application assertion 408, 490
application object 68
ARM EXPRESS-G (figure G.39) 2406
mapping table 680
Angle_taper
application assertion 408, 424, 504, 538, 562
application object 69
ARM EXPRESS-G (figure G.29) 2396
mapping table 681
angular_dimension
AIM expanded listing entities 2006
AIM EXPRESS-G entity (figure H.106) 2525
mapping table 621
angular_location
AIM expanded listing entities 2007
AIM EXPRESS-G entity (figure H.12) 2431
mapping table 1426
Angular_location_dimension
application object 70
ARM EXPRESS-G (figure G.47) 2414
mapping table 1426
angular_size
AIM expanded listing entities 2007
AIM EXPRESS-G entity (figure H.110) 2529
mapping table 1427
Angular_size_dimension
application object 71
ARM EXPRESS-G (figure G.47) 2414
mapping table 1427
Angularity_tolerance
application assertion 408

application object 73
ARM EXPRESS-G (figure G.48) 2415
mapping table 1445
angularity_tolerance
AIM expanded listing entities 2007
AIM EXPRESS-G entity (figure H.110) 2529
mapping table 1445
Annotation_curve
application assertion 418–420, 462, 489, 500, 517, 573
application object 74
ARM EXPRESS-G (figure G.34) 2401
mapping table 621, 1004
annotation_curve_occurrence
AIM expanded listing entities 2007
AIM EXPRESS-G entity (figure H.85) 2504
mapping table 621, 1004
Annotation_element
application assertion 417, 590
application object 74
ARM EXPRESS-G (figure G.34) 2401
mapping table 1004
annotation_fill_area
AIM expanded listing entities 2007
AIM EXPRESS-G entity (figure H.88) 2507
annotation_fill_area_occurrence
AIM expanded listing entities 2007
AIM EXPRESS-G entity (figure H.85) 2504
mapping table 640, 1007, 1008
Annotation_model
application assertion 408
application object 74
ARM EXPRESS-G (figure G.25) 2392
mapping table 1004
annotation_occurrence
AIM expanded listing entities 2007
AIM EXPRESS-G entity (figure H.85) 2504
mapping table 1004
annotation_occurrence_associativity
AIM expanded listing entities 2007
AIM EXPRESS-G entity (figure H.85) 2504
annotation_occurrence_relationship
AIM expanded listing entities 2008
AIM EXPRESS-G entity (figure H.85) 2504
Annotation_subfigure
application assertion 418, 420, 421
application object 75
ARM EXPRESS-G (figure G.34) 2401
mapping table 1004
annotation_subfigure_occurrence
AIM expanded listing entities 2008
AIM EXPRESS-G entity (figure H.85) 2504
mapping table 1004

Annotation_symbol
application assertion 418, 420, 421, 462, 517
application object 75
ARM EXPRESS-G (figure G.33) 2400
mapping table 1005
annotation_symbol
AIM expanded listing entities 2008
AIM EXPRESS-G entity (figure H.17) 2436
annotation_symbol_occurrence
AIM expanded listing entities 2008
AIM EXPRESS-G entity (figure H.85) 2504
mapping table 1005
annotation_text
AIM expanded listing entities 2008
AIM EXPRESS-G entity (figure H.17) 2436
annotation_text_occurrence
AIM expanded listing entities 2009
AIM EXPRESS-G entity (figure H.85) 2504
mapping table 1009
apex
AIM expanded listing entities 2009
AIM EXPRESS-G entity (figure H.54) 2473
Appearance_assignment
application assertion 409, 410, 417, 503, 578, 590
application object 78
ARM EXPRESS-G (figure G.31) 2398
mapping table 984
Application_context
application assertion 410, 461, 546, 547
application object 78
ARM EXPRESS-G (figure G.5) 2372
mapping table 1060
application_context
AIM expanded listing entities 2009
AIM EXPRESS-G entity (figure H.1) 2420
AIM short listing imported entity modifications 1738
mapping table 1060, 1061
application_context_element
AIM expanded listing entities 2009
AIM EXPRESS-G entity (figure H.1) 2420
application_context_element_relationship
AIM expanded listing entities 2009
AIM EXPRESS-G entity (figure H.1) 2420
application_context_requires_ap_definition
AIM expanded listing rules 2255
AIM short listing rules 1769
application_protocol_definition
AIM expanded listing entities 2009
AIM EXPRESS-G entity (figure H.1) 2420
applied_action_assignment
AIM expanded listing entities 2010
AIM EXPRESS-G entity (figure H.41) 2460

AIM short listing entities 1520
mapping table 1274
applied_action_request_assignment
AIM expanded listing entities 2010
AIM EXPRESS-G entity (figure H.39) 2458
AIM short listing entities 1521
applied_approval_assignment
AIM expanded listing entities 2010
AIM EXPRESS-G entity (figure H.21) 2440
AIM short listing entities 1521
applied_certification_assignment
AIM expanded listing entities 2010
AIM EXPRESS-G entity (figure H.35) 2454
AIM short listing entities 1521
applied_contract_assignment
AIM expanded listing entities 2010
AIM EXPRESS-G entity (figure H.35) 2454
AIM short listing entities 1522
applied_date_and_organization_assignment
AIM expanded listing entities 2010
AIM EXPRESS-G entity (figure H.24) 2443
AIM short listing entities 1522
mapping table 1076
applied_date_and_person_and_organization_assignmen
AIM EXPRESS-G entity (figure H.23) 2442
applied_date_and_person_and_organization_assignment
AIM expanded listing entities 2010
AIM short listing entities 1523
mapping table 1076
applied_date_and_person_assignment
AIM expanded listing entities 2010
AIM EXPRESS-G entity (figure H.23) 2442
AIM short listing entities 1523
mapping table 1076
applied_date_and_time_assignment
AIM expanded listing entities 2010
AIM EXPRESS-G entity (figure H.31) 2450
AIM short listing entities 1523
mapping table 1104
applied_date_assignment
AIM expanded listing entities 2011
AIM EXPRESS-G entity (figure H.31) 2450
AIM short listing entities 1524
mapping table 1104
applied_document_reference
AIM expanded listing entities 2011
AIM EXPRESS-G entity (figure H.49) 2468
AIM short listing entities 1524
mapping table 658, 1433
applied_effective_context_assignment
AIM expanded listing entities 2011
AIM EXPRESS-G entity (figure H.36) 2455

AIM short listing entities 1525
applied_effective_item_assignment
AIM expanded listing entities 2011
AIM EXPRESS-G entity (figure H.36) 2455
AIM short listing entities 1525
applied_external_identification_assignment
AIM expanded listing entities 2011
AIM EXPRESS-G entity (figure H.8) 2427
AIM short listing entities 1526
applied_group_assignment
AIM expanded listing entities 2011
AIM EXPRESS-G entity (figure H.52) 2471
AIM short listing entities 1527
applied_identification_assignment
AIM expanded listing entities 2011
AIM EXPRESS-G entity (figure H.8) 2427
AIM short listing entities 1527
mapping table 1048
applied_multi_language_attribute_assignment
AIM expanded listing entities 2011
AIM EXPRESS-G entity (figure H.14) 2433
AIM short listing entities 1527
mapping table 1123
applied_organization_assignment
AIM expanded listing entities 2012
AIM EXPRESS-G entity (figure H.24) 2443
AIM short listing entities 1528
mapping table 1125, 1298
applied_person_and_organization_assignment
AIM expanded listing entities 2012
AIM EXPRESS-G entity (figure H.23) 2442
AIM short listing entities 1528
mapping table 1125
applied_person_assignment
AIM expanded listing entities 2012
AIM EXPRESS-G entity (figure H.23) 2442
AIM short listing entities 1529
mapping table 1125
applied_presented_item
AIM expanded listing entities 2012
AIM EXPRESS-G entity (figure H.90) 2509
AIM short listing entities 1529
mapping table 628, 633
applied_security_classification_assignment
AIM expanded listing entities 2012
AIM EXPRESS-G entity (figure H.34) 2453
AIM short listing entities 1529
Approval
application assertion 410–416, 481
application object 79
ARM EXPRESS-G (figure G.3) 2370
mapping table 1061

approval
AIM expanded listing entities 2012
AIM EXPRESS-G entity (figure H.21) 2440
AIM short listing imported entity modifications 1738
mapping table 1061
approval_assignment
AIM expanded listing entities 2012
AIM EXPRESS-G entity (figure H.21) 2440
approval_date_time
AIM expanded listing entities 2012
AIM EXPRESS-G entity (figure H.21) 2440
approval_date_time_role
AIM expanded listing entities 2012
AIM EXPRESS-G entity (figure H.21) 2440
AIM short listing imported entity modifications 1738
approval_item
AIM expanded listing types 1981
AIM EXPRESS-G types (figure H.22) 2441
AIM short listing types 1498
approval_person_organization
AIM expanded listing entities 2013
AIM EXPRESS-G entity (figure H.21) 2440
AIM short listing imported entity modifications 1739
mapping table 1102
approval_person_organization_requires_date_time
AIM expanded listing rules 2255
AIM short listing rules 1770
mapping table 1074
Approval_relationship
application assertion 416
application object 80
ARM EXPRESS-G (figure G.3) 2370
mapping table 1075
approval_relationship
AIM expanded listing entities 2013
AIM EXPRESS-G entity (figure H.21) 2440
mapping table 1075
approval_requires_approval_assignment
AIM expanded listing rules 2255
AIM short listing rules 1770
approval_role
AIM expanded listing entities 2013
AIM EXPRESS-G entity (figure H.21) 2440
AIM short listing imported entity modifications 1739
Approval_status
application assertion 405, 410, 416
application object 81
ARM EXPRESS-G (figure G.3) 2370
mapping table 1075
approval_status
AIM expanded listing entities 2013
AIM EXPRESS-G entity (figure H.21) 2440

AIM short listing imported entity modifications 1739
approver_role
AIM expanded listing entities 2013
AIM EXPRESS-G entity (figure H.21) 2440
AIM short listing imported entity modifications 1739
Archived_document
application object 81
ARM EXPRESS-G (figure G.10) 2377
mapping table 650
area_in_set
AIM expanded listing entities 2013
AIM EXPRESS-G entity (figure H.90) 2509
area_measure
AIM expanded listing types 1981
AIM EXPRESS-G types (figure H.48) 2467
area_measure_with_unit
AIM expanded listing entities 2013
AIM EXPRESS-G entity (figure H.46) 2465
area_or_view
AIM expanded listing types 1981
AIM EXPRESS-G types (figure H.91) 2510
Area_placed_annotation
application assertion 417
application object 82
ARM EXPRESS-G (figure G.32) 2399
mapping table 1005
area_presentation_relationship
AIM expanded listing entities 2013
AIM short listing entities 1530
mapping table 1010
area_unit
AIM expanded listing entities 2014
AIM EXPRESS-G entity (figure H.47) 2466
aspect_ratio
AIM expanded listing functions 2209
Assembly_component_relationship
application assertion 417, 418, 541, 577
application object 83
ARM EXPRESS-G (figure G.13) 2380
mapping table 1184
assembly_component_usage
AIM expanded listing entities 2014
AIM EXPRESS-G entity (figure H.6) 2425
AIM short listing imported entity modifications 1740
mapping table 1184, 1199, 1204–1206, 1358
assembly_component_usage_substitute
AIM expanded listing entities 2014
AIM EXPRESS-G entity (figure H.6) 2425
AIM short listing imported entity modifications 1740
mapping table 1190
Assembly_definition
application assertion 417

application object ... 83
ARM EXPRESS-G (figure G.13) ... 2380
mapping table ... 1190
assembly_shape_is_defined
AIM expanded listing functions ... 2209
AIM short listing functions ... 1935
Assembly_substitute_relationship
application assertion ... 399, 411, 418, 442, 449, 468, 475, 530, 554, 564, 592
application object ... 84
ARM EXPRESS-G (figure G.13) ... 2380
mapping table ... 1190
Associated_annotation_symbol
application assertion ... 418, 419
application object ... 85
ARM EXPRESS-G (figure G.33) ... 2400
mapping table ... 636
Associated_fill_area
application assertion ... 419
application object ... 85
ARM EXPRESS-G (figure G.34) ... 2401
mapping table ... 640
Associated_leader
application assertion ... 419, 420
application object ... 85
ARM EXPRESS-G (figure G.34) ... 2401
mapping table ... 641
Associated_projection_curve
application assertion ... 420, 421
application object ... 86
ARM EXPRESS-G (figure G.34) ... 2401
mapping table ... 644
associated_surface
AIM expanded listing functions ... 2210
attribute_type
AIM expanded listing types ... 1981
AIM EXPRESS-G types (figure H.14) ... 2433
auto_design_drawing_revision
AIM expanded listing entities ... 2014
AIM EXPRESS-G entity (figure H.107) ... 2526
AIM short listing entities ... 1530
mapping table ... 626
axis1_placement
AIM expanded listing entities ... 2016
AIM EXPRESS-G entity (figure H.59) ... 2478
mapping table ... 1459
axis2_placement
AIM expanded listing types ... 1981
AIM EXPRESS-G types (figure H.62) ... 2481
axis2_placement_2d
AIM expanded listing entities ... 2016
AIM EXPRESS-G entity (figure H.59) ... 2478
mapping table ... 859

axis2_placement_3d
AIM expanded listing entities 2016
AIM EXPRESS-G entity (figure H.59) 2478
mapping table 866, 873, 890, 899, 916, 939, 962, 981–983, 1459
Axis_placement
application assertion 428, 437, 488, 494, 496, 517, 538, 548, 583
application object 86
ARM EXPRESS-G (figure G.40) 2407
mapping table 1163

B_rep
application assertion 421, 422
application object 86
ARM EXPRESS-G (figure G.45) 2412
mapping table 898, 934
B_rep_model
application assertion 422, 440, 574
application object 86
ARM EXPRESS-G (figure G.45) 2412
mapping table 898, 934
B_spline_curve
application assertion 527
application object 87
ARM EXPRESS-G (figure G.43) 2410
mapping table 859, 866, 873, 898, 915, 934, 962
b_spline_curve
AIM expanded listing entities 2016
AIM EXPRESS-G entity (figure H.64) 2483
mapping table 915
b_spline_curve_form
AIM expanded listing types 1982
AIM EXPRESS-G types (figure H.64) 2483
b_spline_curve_with_knots
AIM expanded listing entities 2017
AIM EXPRESS-G entity (figure H.64) 2483
B_spline_surface
application object 87
ARM EXPRESS-G (figure G.44) 2411
mapping table 873, 898, 915, 934, 962
b_spline_surface
AIM expanded listing entities 2017
AIM EXPRESS-G entity (figure H.72) 2491
mapping table 915
b_spline_surface_form
AIM expanded listing types 1982
AIM EXPRESS-G types (figure H.72) 2491
b_spline_surface_with_knots
AIM expanded listing entities 2018
AIM EXPRESS-G entity (figure H.72) 2491
background_colour
AIM expanded listing entities 2018
AIM EXPRESS-G entity (figure H.91) 2510

bag_to_set
AIM expanded listing functions . . . 2210
base_axis
AIM expanded listing functions . . . 2210
basis_curve_check
AIM expanded listing functions . . . 2211
basis_surface_check
AIM expanded listing functions . . . 2213
bead
AIM expanded listing entities . . . 2018
AIM EXPRESS-G entity (figure H.108) . . . 2527
AIM short listing entities . . . 1534
mapping table . . . 683, 791
bead_end
AIM expanded listing entities . . . 2020
AIM EXPRESS-G entity (figure H.11) . . . 2430
AIM short listing entities . . . 1538
mapping table . . . 682, 770, 771, 778, 794
Bead_end_type
application assertion . . . 422
application object . . . 87
ARM EXPRESS-G (figure G.38) . . . 2405
mapping table . . . 682
Bead_feature
application assertion . . . 422
application object . . . 88
ARM EXPRESS-G (figure G.38) . . . 2405
mapping table . . . 683
bezier_curve
AIM expanded listing entities . . . 2021
AIM EXPRESS-G entity (figure H.64) . . . 2483
bezier_surface
AIM expanded listing entities . . . 2021
AIM EXPRESS-G entity (figure H.72) . . . 2491
Block
application object . . . 90
ARM EXPRESS-G (figure G.46) . . . 2413
mapping table . . . 934
block
AIM expanded listing entities . . . 2021
AIM EXPRESS-G entity (figure H.76) . . . 2495
mapping table . . . 945
boolean_choose
AIM expanded listing functions . . . 2213
boolean_operand
AIM expanded listing types . . . 1982
AIM EXPRESS-G types (figure H.76) . . . 2495
boolean_operator
AIM expanded listing types . . . 1982
AIM EXPRESS-G types (figure H.76) . . . 2495
Boolean_result
application assertion . . . 422–424, 440, 574

application object 90
ARM EXPRESS-G (figure G.46) 2413
mapping table 934
boolean_result
AIM expanded listing entities 2021
AIM EXPRESS-G entity (figure H.76) 2495
mapping table 934
boss
AIM expanded listing entities 2021
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1539
mapping table 688, 698, 754, 792, 795
Boss_feature
application assertion 424, 425
application object 91
ARM EXPRESS-G (figure G.37) 2404
mapping table 688
boss_top
AIM expanded listing entities 2023
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1544
mapping table 694, 761, 784
Boss_top_condition
application assertion 424
application object 92
ARM EXPRESS-G (figure G.37) 2404
mapping table 694
boundary_curve
AIM expanded listing entities 2024
AIM EXPRESS-G entity (figure H.65) 2484
bounded_curve
AIM expanded listing entities 2024
AIM EXPRESS-G entity (figure H.61) 2480
bounded_surface
AIM expanded listing entities 2024
AIM EXPRESS-G entity (figure H.69) 2488
box_characteristic_select
AIM expanded listing types 1982
AIM EXPRESS-G types (figure H.100) 2519
box_domain
AIM expanded listing entities 2024
AIM EXPRESS-G entity (figure H.78) 2497
box_height
AIM expanded listing types 1982
AIM EXPRESS-G types (figure H.100) 2519
box_rotate_angle
AIM expanded listing types 1982
AIM EXPRESS-G types (figure H.100) 2519
box_slant_angle
AIM expanded listing types 1982
AIM EXPRESS-G types (figure H.100) 2519
box_width

AIM expanded listing types 1982
AIM EXPRESS-G types (figure H.100) 2519
boxed_half_space
AIM expanded listing entities 2025
AIM EXPRESS-G entity (figure H.78) 2497
brep_with_voids
AIM expanded listing entities 2025
AIM EXPRESS-G entity (figure H.75) 2494
mapping table 953
build_2axes
AIM expanded listing functions 2213
build_axes
AIM expanded listing functions 2214

calendar_date
AIM expanded listing entities 2025
AIM EXPRESS-G entity (figure H.31) 2450
mapping table 1102–1104
camera_image
AIM expanded listing entities 2025
AIM EXPRESS-G entity (figure H.17) 2436
AIM short listing imported entity modifications 1740
camera_image_2d_with_scale
AIM expanded listing entities 2025
AIM EXPRESS-G entity (figure H.17) 2436
AIM short listing entities 1546
mapping table 987, 988
camera_image_3d_with_scale
AIM expanded listing entities 2025
AIM EXPRESS-G entity (figure H.17) 2436
AIM short listing imported entity modifications 1740
mapping table 987–989
camera_model
AIM expanded listing entities 2026
AIM EXPRESS-G entity (figure H.92) 2511
AIM short listing imported entity modifications 1740
camera_model_d2
AIM expanded listing entities 2026
AIM EXPRESS-G entity (figure H.92) 2511
mapping table 987
camera_model_d3
AIM expanded listing entities 2026
AIM EXPRESS-G entity (figure H.92) 2511
mapping table 989
camera_model_d3_with_hlhsr
AIM expanded listing entities 2026
AIM EXPRESS-G entity (figure H.92) 2511
mapping table 1015, 1016
camera_usage
AIM expanded listing entities 2027
AIM EXPRESS-G entity (figure H.17) 2436
Cartesian_coordinate_space

application assertion 408, 425, 487, 498, 524, 525, 529, 538, 540, 549
application object 93
ARM EXPRESS-G (figure G.33) 2400
mapping table 1163
Cartesian_coordinate_space_2d
application assertion 467, 587
application object 93
ARM EXPRESS-G (figure G.33) 2400
mapping table 1164
Cartesian_coordinate_space_3d
application assertion 489
application object 93
ARM EXPRESS-G (figure G.33) 2400
mapping table 1164
Cartesian_point
application assertion 539, 540
application object 93
ARM EXPRESS-G (figure G.43) 2410
mapping table 859, 866, 873, 890, 898, 915, 938, 962
cartesian_point
AIM expanded listing entities 2027
AIM EXPRESS-G entity (figure H.58) 2477
mapping table 859, 860, 865–867, 872–875, 888, 890, 899–901, 915–917, 932, 939–942, 962–964, 968
cartesian_transformation_operator
AIM expanded listing entities 2027
AIM EXPRESS-G entity (figure H.60) 2479
mapping table 1180
cartesian_transformation_operator_2d
AIM expanded listing entities 2027
AIM EXPRESS-G entity (figure H.60) 2479
cartesian_transformation_operator_3d
AIM expanded listing entities 2027
AIM EXPRESS-G entity (figure H.60) 2479
mapping table 981
Catalog_thread
application assertion 425
application object 94
ARM EXPRESS-G (figure G.39) 2406
mapping table 694
category_usage_item
AIM expanded listing types 1982
AIM EXPRESS-G types (figure H.52) 2471
AIM short listing types 1499
central_or_parallel
AIM expanded listing types 1982
AIM EXPRESS-G types (figure H.92) 2511
Centre_of_mass
application assertion 425
application object 94
ARM EXPRESS-G (figure G.24) 2391
mapping table 1021

centre_of_symmetry
AIM expanded listing entities ... 2028
AIM EXPRESS-G entity (figure H.54) ... 2473
certification
AIM expanded listing entities ... 2028
AIM EXPRESS-G entity (figure H.35) ... 2454
AIM short listing imported entity modifications ... 1741
mapping table ... 1374
certification_assignment
AIM expanded listing entities ... 2028
AIM EXPRESS-G entity (figure H.35) ... 2454
certification_item
AIM expanded listing types ... 1982
AIM EXPRESS-G types (figure H.35) ... 2454
AIM short listing types ... 1500
certification_role
AIM expanded listing entities ... 2028
AIM EXPRESS-G entity (figure H.35) ... 2454
AIM short listing imported entity modifications ... 1741
certification_type
AIM expanded listing entities ... 2028
AIM EXPRESS-G entity (figure H.35) ... 2454
mapping table ... 1374
Chamfer
application assertion ... 425
application object ... 95
ARM EXPRESS-G (figure G.29) ... 2396
mapping table ... 695
chamfer
AIM expanded listing entities ... 2028
AIM EXPRESS-G entity (figure H.108) ... 2527
AIM short listing entities ... 1547
mapping table ... 695
Chamfer_angle
application assertion ... 426
application object ... 95
ARM EXPRESS-G (figure G.29) ... 2396
mapping table ... 695
Chamfer_definition
application assertion ... 425, 426, 490, 562
application object ... 96
ARM EXPRESS-G (figure G.29) ... 2396
mapping table ... 696
Chamfer_offset
application assertion ... 426
application object ... 97
ARM EXPRESS-G (figure G.29) ... 2396
mapping table ... 697
chamfer_offset
AIM expanded listing entities ... 2029
AIM EXPRESS-G entity (figure H.11) ... 2430
AIM short listing entities ... 1548

mapping table 695, 697
Change
application assertion 426–428, 482, 524, 550
application object 97
ARM EXPRESS-G (figure G.18) 2385
mapping table 1290
character_spacing_select
AIM expanded listing types 1983
AIM EXPRESS-G types (figure H.104) 2523
character_style_select
AIM expanded listing types 1983
AIM EXPRESS-G types (figure H.100) 2519
characterized_action_definition
AIM expanded listing types 1983
AIM EXPRESS-G types (figure H.102) 2521
characterized_definition
AIM expanded listing types 1983
AIM EXPRESS-G types (figure H.10) 2429
characterized_material_property
AIM expanded listing types 1983
AIM EXPRESS-G types (figure H.56) 2475
characterized_object
AIM expanded listing entities 2030
AIM EXPRESS-G entity (figure H.2) 2421
AIM short listing imported entity modifications 1741
mapping table 711, 717, 749, 763, 1362
characterized_object_relationship
AIM expanded listing entities 2030
AIM EXPRESS-G entity (figure H.2) 2421
mapping table 809
characterized_object_subtype_exclusiveness
AIM expanded listing rules 2255
AIM short listing rules 1771
characterized_product_category
AIM expanded listing entities 2030
AIM EXPRESS-G entity (figure H.2) 2421
AIM short listing entities 1550
mapping table 1333
characterized_product_definition
AIM expanded listing types 1983
AIM EXPRESS-G types (figure H.10) 2429
characterized_resource_definition
AIM expanded listing types 1983
AIM EXPRESS-G types (figure H.40) 2459
check_associative_shape_aspects
AIM expanded listing functions 2214
check_text_alignment
AIM expanded listing functions 2215
check_text_font
AIM expanded listing functions 2215
Circle
application assertion 527

application object 98
ARM EXPRESS-G (figure G.43) 2410
mapping table 859, 866, 873, 898, 915, 938, 962
circle
AIM expanded listing entities 2030
AIM EXPRESS-G entity (figure H.62) 2481
mapping table 915
Circular_boss
application assertion 428
application object 98
ARM EXPRESS-G (figure G.37) 2404
mapping table 698
Circular_closed_profile
application assertion 428, 562
application object 98
ARM EXPRESS-G (figure G.30) 2397
mapping table 698
circular_closed_profile
AIM expanded listing entities 2030
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1551
mapping table 698, 708, 792
Circular_offset_pattern
application assertion 428
application object 99
ARM EXPRESS-G (figure G.40) 2407
mapping table 699
Circular_omit_pattern
application assertion 428
application object 99
ARM EXPRESS-G (figure G.40) 2407
mapping table 701
Circular_path
application assertion 428
application object 101
ARM EXPRESS-G (figure G.29) 2396
mapping table 702
Circular_pattern
application assertion 428
application object 101
ARM EXPRESS-G (figure G.40) 2407
mapping table 704
circular_pattern
AIM expanded listing entities 2031
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1553
mapping table 704
Circular_runout_tolerance
application assertion 429
application object 103
ARM EXPRESS-G (figure G.48) 2415
mapping table 1445

circular_runout_tolerance
AIM expanded listing entities 2032
AIM EXPRESS-G entity (figure H.110) 2529
mapping table 1445
Circularity_tolerance
application object 104
ARM EXPRESS-G (figure G.48) 2415
mapping table 1446
circularity_tolerance
AIM expanded listing entities 2032
AIM EXPRESS-G entity (figure H.109) 2528
mapping table 1446
Class_category_association
application assertion 399, 411, 429, 442, 450, 468, 475, 530, 546, 554, 592
application object 104
ARM EXPRESS-G (figure G.6) 2373
mapping table 1337
Class_condition_association
application assertion 399, 411, 429, 436, 443, 450, 468, 475, 530, 555, 592
application object 105
ARM EXPRESS-G (figure G.6) 2373
mapping table 1337
Class_function_association
application assertion 399, 411, 429, 430, 443, 450, 465, 468, 476, 482, 531, 555, 564, 588, 593
application object 106
ARM EXPRESS-G (figure G.7) 2374
mapping table 1338
Class_inclusion_association
application assertion 400, 411, 430, 443, 450, 469, 476, 531, 555, 593
application object 107
ARM EXPRESS-G (figure G.6) 2373
mapping table 1339
Class_specification_association
application assertion 400, 411, 430, 436, 443, 450, 469, 476, 531, 555, 593
application object 108
ARM EXPRESS-G (figure G.6) 2373
mapping table 1340
Classification_attribute
application assertion 431, 492
application object 109
ARM EXPRESS-G (figure G.1) 2368
mapping table 1329
Classification_system
application assertion 405, 416, 431, 432, 464, 492, 568
application object 111
ARM EXPRESS-G (figure G.1) 2368
mapping table 1331
classification_system
AIM expanded listing entities 2032
AIM EXPRESS-G entity (figure H.52) 2471
AIM short listing entities 1555
mapping table 1331

closed_path_profile
AIM expanded listing entities 2032
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1556
mapping table 708, 756, 757, 792
Closed_profile
application assertion 432
application object 112
ARM EXPRESS-G (figure G.30) 2397
mapping table 708
Closed_profile_usage
application assertion 432, 491, 504, 528
application object 112
ARM EXPRESS-G (figure G.39) 2406
mapping table 709
Closed_shell
application assertion 421, 432
application object 112
ARM EXPRESS-G (figure G.45) 2412
mapping table 898, 939
closed_shell
AIM expanded listing entities 2033
AIM EXPRESS-G entity (figure H.83) 2502
mapping table 888, 893, 952
Coating_layer
application assertion 432
application object 113
ARM EXPRESS-G (figure G.50) 2417
mapping table 602
Coating_layer_relationship
application assertion 432, 433
application object 113
ARM EXPRESS-G (figure G.50) 2417
mapping table 603
Collected_item_association
application assertion 433
application object 115
ARM EXPRESS-G (figure G.13) 2380
mapping table 1341
Collection_definition
application assertion 433
application object 115
ARM EXPRESS-G (figure G.13) 2380
mapping table 1342
colour
AIM expanded listing entities 2033
AIM EXPRESS-G entity (figure H.91) 2510
colour_rgb
AIM expanded listing entities 2033
AIM EXPRESS-G entity (figure H.91) 2510
AIM short listing imported entity modifications 1741
colour_specification

AIM expanded listing entities 2033
AIM EXPRESS-G entity (figure H.91) 2510
Complete_circular_path
application object 115
ARM EXPRESS-G (figure G.29) 2396
mapping table 711
component_function_definition
AIM expanded listing entities 2033
AIM short listing entities 1557
mapping table 1351
Composite_curve
application object 115
ARM EXPRESS-G (figure G.43) 2410
mapping table 859, 866, 962
composite_curve
AIM expanded listing entities 2034
AIM EXPRESS-G entity (figure H.65) 2484
composite_curve_on_surface
AIM expanded listing entities 2034
AIM EXPRESS-G entity (figure H.65) 2484
composite_curve_segment
AIM expanded listing entities 2034
AIM EXPRESS-G entity (figure H.65) 2484
composite_shape_aspect
AIM expanded listing entities 2034
AIM EXPRESS-G entity (figure H.11) 2430
mapping table 672
composite_text
AIM expanded listing entities 2034
AIM EXPRESS-G entity (figure H.87) 2506
composite_text_with_associated_curves
AIM expanded listing entities 2035
AIM EXPRESS-G entity (figure H.87) 2506
composite_text_with_blanking_box
AIM expanded listing entities 2035
AIM EXPRESS-G entity (figure H.87) 2506
composite_text_with_extent
AIM expanded listing entities 2035
AIM EXPRESS-G entity (figure H.87) 2506
Compound_datum
application assertion 433
application object 116
ARM EXPRESS-G (figure G.49) 2416
mapping table 1446
Compound_feature_in_panel
application assertion 433
application object 116
ARM EXPRESS-G (figure G.28) 2395
mapping table 711
compound_feature_in_panel
AIM expanded listing entities 2035
AIM EXPRESS-G entity (figure H.108) 2527

AIM short listing entities 1557
mapping table 711, 791
Compound_feature_in_panel_element
application assertion 433, 434
application object 117
ARM EXPRESS-G (figure G.28) 2395
mapping table 712
Compound_feature_in_panel_element_relationship
application assertion 434
application object 117
ARM EXPRESS-G (figure G.28) 2395
mapping table 717
Compound_feature_in_solid
application assertion 434
application object 118
ARM EXPRESS-G (figure G.28) 2395
mapping table 717
compound_feature_in_solid
AIM expanded listing entities 2036
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1560
mapping table 717, 792
Compound_feature_in_solid_element
application assertion 434, 435
application object 118
ARM EXPRESS-G (figure G.28) 2395
mapping table 717
Compound_feature_in_solid_element_relationship
application assertion 435
application object 119
ARM EXPRESS-G (figure G.28) 2395
mapping table 723
compound_item_definition
AIM expanded listing types 1983
AIM EXPRESS-G types (figure H.17) 2436
Compound_model
application assertion 435
application object 120
ARM EXPRESS-G (figure G.45) 2412
mapping table 915
compound_representation_item
AIM expanded listing entities 2038
AIM EXPRESS-G entity (figure H.17) 2436
mapping table 982, 983
compound_shape_representation
AIM expanded listing entities 2038
AIM EXPRESS-G entity (figure H.20) 2439
AIM short listing entities 1563
mapping table 915
Concentricity_tolerance
application assertion 435
application object 120

ARM EXPRESS-G (figure G.48) 2415
mapping table 1446
concentricity_tolerance
AIM expanded listing entities 2039
AIM EXPRESS-G entity (figure H.110) 2529
mapping table 1446
concept_feature_relationship
AIM expanded listing entities 2039
AIM EXPRESS-G entity (figure H.7) 2426
concept_feature_relationship_with_condition
AIM expanded listing entities 2039
AIM EXPRESS-G entity (figure H.7) 2426
mapping table 1379
conditional_concept_feature
AIM expanded listing entities 2039
AIM EXPRESS-G entity (figure H.7) 2426
mapping table 1378, 1380
conditional_reverse
AIM expanded listing functions 2215
configurable_item
AIM expanded listing entities 2039
AIM EXPRESS-G entity (figure H.3) 2422
AIM short listing imported entity modifications 1741
mapping table 1370
Configuration
application assertion 400, 412, 436, 443, 450, 476, 531, 555, 564, 593
application object 121
ARM EXPRESS-G (figure G.8) 2375
mapping table 1343
configuration_definition
AIM expanded listing entities 2039
AIM EXPRESS-G entity (figure H.123) 2542
mapping table 971
configuration_design
AIM expanded listing entities 2039
AIM EXPRESS-G entity (figure H.3) 2422
AIM short listing imported entity modifications 1742
mapping table 1338
configuration_design_item
AIM expanded listing types 1983
AIM EXPRESS-G types (figure H.4) 2423
configuration_design_subtype_exclusiveness
AIM expanded listing rules 2255
AIM short listing rules 1772
mapping table 1201, 1351, 1355
configuration_interpolation
AIM expanded listing entities 2040
AIM EXPRESS-G entity (figure H.123) 2542
mapping table 971, 972
configuration_item
AIM expanded listing entities 2040
AIM EXPRESS-G entity (figure H.3) 2422

mapping table 1364, 1371
configuration_item_context
AIM expanded listing entities 2040
AIM EXPRESS-G entity (figure H.1) 2420
mapping table 1060, 1061
configuration_item_relationship
AIM expanded listing entities 2040
AIM EXPRESS-G entity (figure H.3) 2422
AIM short listing imported entity modifications 1742
mapping table 1290, 1292, 1366
configuration_item_with_context
AIM expanded listing entities 2040
AIM EXPRESS-G entity (figure H.3) 2422
configured_effective_context_assignment
AIM expanded listing entities 2040
AIM EXPRESS-G entity (figure H.36) 2455
AIM short listing entities 1564
mapping table 1343
configured_effective_context_item
AIM expanded listing types 1983
AIM EXPRESS-G types (figure H.37) 2456
AIM short listing types 1500
configured_effective_item
AIM expanded listing types 1983
AIM EXPRESS-G types (figure H.37) 2456
AIM short listing types 1500
configured_effective_item_assignment
AIM expanded listing entities 2040
AIM EXPRESS-G entity (figure H.36) 2455
AIM short listing entities 1565
mapping table 1343
conic
AIM expanded listing entities 2041
AIM EXPRESS-G entity (figure H.62) 2481
Conical_bottom
application assertion 436
application object 121
ARM EXPRESS-G (figure G.36) 2403
mapping table 723
Conical_surface
application object 122
ARM EXPRESS-G (figure G.44) 2411
mapping table 873, 899, 916, 939, 962
conical_surface
AIM expanded listing entities 2041
AIM EXPRESS-G entity (figure H.70) 2489
mapping table 916
connected_edge_set
AIM expanded listing entities 2041
AIM EXPRESS-G entity (figure H.83) 2502
connected_face_set
AIM expanded listing entities 2041

AIM EXPRESS-G entity (figure H.83) 2502
connected_face_shape_representation
AIM expanded listing entities 2041
AIM EXPRESS-G entity (figure H.16) 2435
AIM short listing entities 1566
Constant_radius_edge_round
application assertion 436
application object 122
ARM EXPRESS-G (figure G.29) 2396
mapping table 725
Constant_radius_fillet
application assertion 437
application object 123
ARM EXPRESS-G (figure G.29) 2396
mapping table 727
constraints_composite_curve_on_surface
AIM expanded listing functions 2216
constraints_geometry_shell_based_surface_model
AIM expanded listing functions 2216
constraints_param_b_spline
AIM expanded listing functions 2216
constraints_rectangular_composite_surface
AIM expanded listing functions 2217
Constructive_geometric_element
application assertion 437, 438, 460, 500, 518, 573
application object 124
ARM EXPRESS-G (figure G.42) 2409
mapping table 859, 866, 873, 890, 899, 916, 939, 962
Constructive_geometry
application assertion 438
application object 125
ARM EXPRESS-G (figure G.42) 2409
mapping table 859, 866, 874, 890, 900, 918, 941, 963
Constructive_geometry_association
application assertion 438
application object 125
ARM EXPRESS-G (figure G.42) 2409
mapping table 860, 867, 875, 891, 901, 919, 942, 963
constructive_geometry_representation
AIM expanded listing entities 2041
AIM EXPRESS-G entity (figure H.16) 2435
AIM short listing entities 1567
mapping table 859, 866, 874, 890, 900, 918, 941, 963
constructive_geometry_representation_relationship
AIM expanded listing entities 2042
AIM EXPRESS-G entity (figure H.18) 2437
AIM short listing entities 1568
mapping table 860, 867, 875, 891, 901, 919, 942, 963
Contact_ratio
application assertion 439
application object 126
ARM EXPRESS-G (figure G.50) 2417

mapping table 605
Context_dependent_appearance_assignment
application assertion 439, 440
application object 127
ARM EXPRESS-G (figure G.31) 2398
mapping table 985
context_dependent_invisibility
AIM expanded listing entities 2042
AIM EXPRESS-G entity (figure H.101) 2520
context_dependent_measure
AIM expanded listing types 1984
AIM EXPRESS-G types (figure H.48) 2467
context_dependent_over_riding_styled_item
AIM expanded listing entities 2042
AIM EXPRESS-G entity (figure H.93) 2512
context_dependent_shape_representation
AIM expanded listing entities 2042
AIM EXPRESS-G entity (figure H.13) 2432
context_dependent_unit
AIM expanded listing entities 2043
AIM EXPRESS-G entity (figure H.47) 2466
Contract
application assertion 440, 482, 526
application object 127
ARM EXPRESS-G (figure G.17) 2384
mapping table 1295
contract
AIM expanded listing entities 2043
AIM EXPRESS-G entity (figure H.35) 2454
contract_assignment
AIM expanded listing entities 2043
AIM EXPRESS-G entity (figure H.35) 2454
contract_item
AIM expanded listing types 1984
AIM EXPRESS-G types (figure H.35) 2454
AIM short listing types 1500
contract_role
AIM expanded listing entities 2043
AIM EXPRESS-G entity (figure H.35) 2454
AIM short listing imported entity modifications 1742
contract_type
AIM expanded listing entities 2043
AIM EXPRESS-G entity (figure H.35) 2454
conversion_based_unit
AIM expanded listing entities 2043
AIM EXPRESS-G entity (figure H.47) 2466
convert_plane_angle_for_pair_from_radian
AIM expanded listing functions 2218
convert_spatial_to_ypr_rotation
AIM expanded listing functions 2219
coordinated_assembly_and_shape
AIM expanded listing rules 2256

AIM short listing rules 1773
mapping table 1203
coordinated_pair_link_representation
AIM expanded listing functions 2222
coordinated_universal_time_offset
AIM expanded listing entities 2043
AIM EXPRESS-G entity (figure H.31) 2450
count_measure
AIM expanded listing types 1984
AIM EXPRESS-G types (figure H.48) 2467
cross_product
AIM expanded listing functions 2223
Csg_model
application assertion 440, 441
application object 128
ARM EXPRESS-G (figure G.46) 2413
mapping table 943
Csg_primitive
application assertion 423, 440, 574
application object 128
ARM EXPRESS-G (figure G.46) 2413
mapping table 945
csg_primitive
AIM expanded listing types 1984
AIM EXPRESS-G types (figure H.74) 2493
csg_select
AIM expanded listing types 1984
AIM EXPRESS-G types (figure H.74) 2493
csg_shape_representation
AIM expanded listing entities 2043
AIM EXPRESS-G entity (figure H.20) 2439
mapping table 943
csg_solid
AIM expanded listing entities 2044
AIM EXPRESS-G entity (figure H.74) 2493
Curve
application assertion 437, 441, 493, 496, 500
application object 128
ARM EXPRESS-G (figure G.43) 2410
mapping table 861, 868, 876, 902, 920, 945, 964
curve
AIM expanded listing entities 2044
AIM EXPRESS-G entity (figure H.61) 2480
mapping table 862–864, 873, 899–901, 916, 920, 939, 941, 942, 962
Curve_appearance
application assertion 409
application object 128
ARM EXPRESS-G (figure G.31) 2398
mapping table 987, 1015
curve_bounded_surface
AIM expanded listing entities 2044
AIM EXPRESS-G entity (figure H.73) 2492

mapping table 969, 971, 972, 975
curve_dimension
AIM expanded listing entities 2044
AIM EXPRESS-G entity (figure H.106) 2525
mapping table 621, 622
curve_font_or_scaled_curve_font_select
AIM expanded listing types 1984
AIM EXPRESS-G types (figure H.96) 2515
curve_font_usage
AIM expanded listing rules 2256
AIM short listing rules 1773
mapping table 987, 997
Curve_on_surface
application assertion 527, 561
application object 129
ARM EXPRESS-G (figure G.43) 2410
mapping table 876, 902, 920, 945, 964
curve_on_surface
AIM expanded listing types 1984
AIM EXPRESS-G types (figure H.68) 2487
curve_or_annotation_curve_occurrence
AIM expanded listing types 1984
AIM EXPRESS-G types (figure H.104) 2523
curve_or_render
AIM expanded listing types 1984
AIM EXPRESS-G types (figure H.95) 2514
curve_replica
AIM expanded listing entities 2044
AIM EXPRESS-G entity (figure H.67) 2486
Curve_set_2d
application assertion 441, 591
application object 130
ARM EXPRESS-G (figure G.43) 2410
mapping table 861
Curve_set_3d
application assertion 441, 442, 591
application object 130
ARM EXPRESS-G (figure G.43) 2410
mapping table 868
curve_style
AIM expanded listing entities 2045
AIM EXPRESS-G entity (figure H.96) 2515
AIM short listing imported entity modifications 1742
mapping table 621, 987, 997, 1004, 1015
curve_style_font
AIM expanded listing entities 2045
AIM EXPRESS-G entity (figure H.96) 2515
AIM short listing imported entity modifications 1743
curve_style_font_pattern
AIM expanded listing entities 2045
AIM EXPRESS-G entity (figure H.96) 2515
curve_style_font_select

AIM expanded listing types 1984
AIM EXPRESS-G types (figure H.96) 2515
curve_style_rendering
AIM expanded listing entities 2045
AIM EXPRESS-G entity (figure H.95) 2514
mapping table 1015
curve_weights_positive
AIM expanded listing functions 2223
Curved_distance_dimension
application assertion 442
application object 130
ARM EXPRESS-G (figure G.47) 2414
mapping table 1427
cylindrical_pair
AIM expanded listing entities 2045
AIM EXPRESS-G entity (figure H.116) 2535
cylindrical_pair_range
AIM expanded listing entities 2045
AIM EXPRESS-G entity (figure H.120) 2539
cylindrical_pair_value
AIM expanded listing entities 2046
AIM EXPRESS-G entity (figure H.112) 2531
Cylindrical_surface
application object 131
ARM EXPRESS-G (figure G.44) 2411
mapping table 876, 902, 920, 945, 965
cylindrical_surface
AIM expanded listing entities 2046
AIM EXPRESS-G entity (figure H.70) 2489
mapping table 876, 902, 920, 945, 965
Cylindricity_tolerance
application object 131
ARM EXPRESS-G (figure G.48) 2415
mapping table 1446
cylindricity_tolerance
AIM expanded listing entities 2046
AIM EXPRESS-G entity (figure H.109) 2528
mapping table 1446

Data_environment
application assertion 442, 523
application object 131
ARM EXPRESS-G (figure G.24) 2391
mapping table 1022
data_environment
AIM expanded listing entities 2046
AIM EXPRESS-G entity (figure H.56) 2475
AIM short listing imported entity modifications 1743
mapping table 609, 610, 1022
date
AIM expanded listing entities 2046
AIM EXPRESS-G entity (figure H.31) 2450

AIM short listing imported entity modifications 1743
date_and_organization_item
AIM expanded listing types 1984
AIM EXPRESS-G types (figure H.28) 2447
AIM short listing types 1501
date_and_person_and_organization_item
AIM expanded listing types 1985
AIM EXPRESS-G types (figure H.26) 2445
AIM short listing types 1502
Date_and_person_assignment
application assertion 442–448
application object 132
ARM EXPRESS-G (figure G.2) 2369
mapping table 1076
date_and_person_item
AIM expanded listing types 1985
AIM EXPRESS-G types (figure H.27) 2446
AIM short listing types 1503
Date_and_person_or_organization
application assertion 397, 412, 444, 448, 449, 526, 593
application object 133
ARM EXPRESS-G (figure G.2) 2369
mapping table 1102
date_and_time
AIM expanded listing entities 2046
AIM EXPRESS-G entity (figure H.31) 2450
AIM short listing imported entity modifications 1743
mapping table 1102
date_and_time_assignment
AIM expanded listing entities 2046
AIM EXPRESS-G entity (figure H.31) 2450
date_and_time_item
AIM expanded listing types 1986
AIM EXPRESS-G types (figure H.33) 2452
AIM short listing types 1504
date_assignment
AIM expanded listing entities 2046
AIM EXPRESS-G entity (figure H.31) 2450
date_item
AIM expanded listing types 1986
AIM EXPRESS-G types (figure H.32) 2451
AIM short listing types 1505
date_role
AIM expanded listing entities 2047
AIM EXPRESS-G entity (figure H.31) 2450
AIM short listing imported entity modifications 1744
mapping table 1104
Date_time
application assertion 398, 412, 448, 451, 469, 549, 555, 575
application object 133
ARM EXPRESS-G (figure G.2) 2369
mapping table 1103

Date_time_assignment
application assertion 449–455
application object 134
ARM EXPRESS-G (figure G.2) 2369
mapping table 1104
date_time_role
AIM expanded listing entities 2047
AIM EXPRESS-G entity (figure H.31) 2450
AIM short listing imported entity modifications 1744
mapping table 1104
date_time_select
AIM expanded listing types 1987
AIM EXPRESS-G types (figure H.21) 2440
Dated_configuration
application assertion 469
application object 135
ARM EXPRESS-G (figure G.8) 2375
mapping table 1345
dated_effectivity
AIM expanded listing entities 2047
AIM EXPRESS-G entity (figure H.36) 2455
mapping table 1209, 1210, 1226, 1345
Datum
application assertion 408, 429, 435, 517, 528, 529, 540, 580, 581, 585
application object 135
ARM EXPRESS-G (figure G.49) 2416
mapping table 1446
datum
AIM expanded listing entities 2047
AIM EXPRESS-G entity (figure H.110) 2529
mapping table 1446, 1447
Datum_feature
application assertion 433, 455, 456
application object 135
ARM EXPRESS-G (figure G.49) 2416
mapping table 1446
datum_feature
AIM expanded listing entities 2047
AIM EXPRESS-G entity (figure H.11) 2430
mapping table 1446
datum_feature_callout
AIM expanded listing entities 2047
AIM EXPRESS-G entity (figure H.106) 2525
datum_reference
AIM expanded listing entities 2048
AIM EXPRESS-G entity (figure H.110) 2529
mapping table 1446
Datum_target
application assertion 456
application object 136
ARM EXPRESS-G (figure G.49) 2416
mapping table 1447

datum_target
AIM expanded listing entities 2048
AIM EXPRESS-G entity (figure H.11) 2430
mapping table 1447
datum_target_callout
AIM expanded listing entities 2048
AIM EXPRESS-G entity (figure H.106) 2525
Datum_target_set
application assertion 455, 456
application object 136
ARM EXPRESS-G (figure G.49) 2416
mapping table 1447
day_in_month_number
AIM expanded listing types 1987
AIM EXPRESS-G types (figure H.31) 2450
Default_chamfer
application assertion 456, 457
application object 137
ARM EXPRESS-G (figure G.26) 2393
mapping table 729
Default_draft_angle
application assertion 456, 457
application object 137
ARM EXPRESS-G (figure G.26) 2393
mapping table 730
Default_edge_round
application assertion 456–458
application object 138
ARM EXPRESS-G (figure G.26) 2393
mapping table 731
Default_fillet
application assertion 457, 458
application object 139
ARM EXPRESS-G (figure G.26) 2393
mapping table 732
Default_roughness
application assertion 457, 458
application object 139
ARM EXPRESS-G (figure G.26) 2393
mapping table 606
Default_setting_association
application assertion 457–459
application object 140
ARM EXPRESS-G (figure G.26) 2393
mapping table 607, 733, 1022, 1428
Default_thickness
application assertion 458, 459
application object 140
ARM EXPRESS-G (figure G.26) 2393
mapping table 1024
default_tolerance_table
AIM expanded listing entities 2048

AIM EXPRESS-G entity (figure H.16) 2435
AIM short listing entities 1569
mapping table 1433
default_tolerance_table_cell
AIM expanded listing entities 2048
AIM EXPRESS-G entity (figure H.16) 2435
AIM short listing entities 1569
mapping table 1441
defined_symbol
AIM expanded listing entities 2049
AIM EXPRESS-G entity (figure H.88) 2507
defined_symbol_select
AIM expanded listing types 1987
AIM EXPRESS-G types (figure H.88) 2507
Defined_thread
application assertion 459
application object 141
ARM EXPRESS-G (figure G.39) 2406
mapping table 735
definition_replacement_requires_effectivity_assignment
AIM expanded listing rules 2256
AIM short listing rules 1774
definitional_representation
AIM expanded listing entities 2049
AIM EXPRESS-G entity (figure H.16) 2435
degenerate_pcurve
AIM expanded listing entities 2049
AIM EXPRESS-G entity (figure H.58) 2477
mapping table 873, 916, 932, 934, 962
degenerate_toroidal_surface
AIM expanded listing entities 2050
AIM EXPRESS-G entity (figure H.70) 2489
mapping table 889, 969
dependent_instantiable_action_request_role
AIM expanded listing rules 2256
AIM short listing rules 1775
mapping table 1313–1327
dependent_instantiable_action_role
AIM expanded listing rules 2256
AIM short listing rules 1776
mapping table 1211, 1226, 1227, 1246, 1274, 1296, 1304, 1305, 1307, 1397, 1401
dependent_instantiable_approval_date_time_role
AIM expanded listing rules 2256
AIM short listing rules 1776
mapping table 1061, 1075
dependent_instantiable_approval_role
AIM expanded listing rules 2257
AIM short listing rules 1777
mapping table 1061
dependent_instantiable_approver_role
AIM expanded listing rules 2257
AIM short listing rules 1778

mapping table 1074
dependent_instantiable_certification_role
AIM expanded listing rules 2257
AIM short listing rules 1778
mapping table 1375
dependent_instantiable_colour_rgb
AIM expanded listing rules 2257
AIM short listing rules 1779
mapping table 987, 990, 1015, 1016
dependent_instantiable_contract_role
AIM expanded listing rules 2257
AIM short listing rules 1779
mapping table 1295, 1296
dependent_instantiable_curve_style
AIM expanded listing rules 2257
AIM short listing rules 1780
mapping table 987, 997, 1015
dependent_instantiable_date
AIM expanded listing rules 2257
AIM short listing rules 1780
dependent_instantiable_date_and_time
AIM expanded listing rules 2257
AIM short listing rules 1781
mapping table 1103
dependent_instantiable_date_role
AIM expanded listing rules 2258
AIM short listing rules 1781
dependent_instantiable_date_time_role
AIM expanded listing rules 2258
AIM short listing rules 1782
dependent_instantiable_derived_unit
AIM expanded listing rules 2258
AIM short listing rules 1783
mapping table 618, 677, 1044, 1207, 1266, 1309
dependent_instantiable_dimensional_size
AIM expanded listing rules 2258
AIM short listing rules 1783
mapping table 1440
dependent_instantiable_document_role
AIM expanded listing rules 2258
AIM short listing rules 1784
mapping table 608, 658, 668, 695, 733, 1022, 1329, 1331, 1334, 1428
dependent_instantiable_effectivity_role
AIM expanded listing rules 2258
AIM short listing rules 1784
mapping table 1209, 1210, 1343
dependent_instantiable_externally_defined_symbol
AIM expanded listing rules 2258
AIM short listing rules 1785
mapping table 1005
dependent_instantiable_fill_area_style
AIM expanded listing rules 2258

AIM short listing rules 1785
mapping table 984, 985, 995
dependent_instantiable_fill_area_style_colour
AIM expanded listing rules 2259
AIM short listing rules 1786
mapping table 1009
dependent_instantiable_group_role
AIM expanded listing rules 2259
AIM short listing rules 1786
mapping table 654, 674, 1190, 1335, 1337, 1342, 1365, 1377, 1393
dependent_instantiable_identification_role
AIM expanded listing rules 2259
AIM short listing rules 1787
mapping table 603, 650, 651, 667, 864, 871, 885, 896, 911, 929, 955, 967, 994, 1001, 1048, 1309, 1349, 1350, 1355, 1362, 1364, 1368, 1371, 1376, 1413
dependent_instantiable_language_role
AIM expanded listing rules 2259
AIM short listing rules 1788
mapping table 1159
dependent_instantiable_measure_with_unit
AIM expanded listing rules 2259
AIM short listing rules 1788
mapping table 618, 677, 1044, 1207, 1266, 1309
dependent_instantiable_named_unit
AIM expanded listing rules 2259
AIM short listing rules 1789
dependent_instantiable_organization_role
AIM expanded listing rules 2259
AIM short listing rules 1789
mapping table 1076, 1125, 1272, 1299, 1311, 1312
dependent_instantiable_person_and_organization_role
AIM expanded listing rules 2259
AIM short listing rules 1790
dependent_instantiable_person_role
AIM expanded listing rules 2260
AIM short listing rules 1790
mapping table 1076, 1125, 1272, 1299, 1311, 1312
dependent_instantiable_pre_defined_colour
AIM expanded listing rules 2260
AIM short listing rules 1791
mapping table 987, 990, 1015, 1016
dependent_instantiable_pre_defined_symbol
AIM expanded listing rules 2260
AIM short listing rules 1791
mapping table 624, 997, 1008
dependent_instantiable_presentation_style_by_context
AIM expanded listing rules 2260
AIM short listing rules 1792
mapping table 985
dependent_instantiable_representation
AIM expanded listing rules 2260
AIM short listing rules 1793

mapping table 668, 862, 869, 883, 894, 909, 927, 952, 965, 991
dependent_instantiable_security_classification_role
AIM expanded listing rules 2260
AIM short listing rules 1793
mapping table 1149–1157
dependent_instantiable_symbol_colour
AIM expanded listing rules 2260
AIM short listing rules 1794
mapping table 1005
dependent_instantiable_text_style
AIM expanded listing rules 2261
AIM short listing rules 1794
mapping table 1010
dependent_instantiable_text_style_for_defined_font
AIM expanded listing rules 2261
AIM short listing rules 1795
mapping table 1010
dependent_instantiable_tolerance_value
AIM expanded listing rules 2261
AIM short listing rules 1795
mapping table 1439
dependent_instantiable_uncertainty_qualifier
AIM expanded listing rules 2261
AIM short listing rules 1796
mapping table 1161, 1162
derive_dimensional_exponents
AIM expanded listing functions 2224
Derived_geometry
application assertion 455, 460, 500, 518
application object 143
ARM EXPRESS-G (figure G.49) 2416
mapping table 1429, 1448
derived_shape_aspect
AIM expanded listing entities 2050
AIM EXPRESS-G entity (figure H.54) 2473
derived_unit
AIM expanded listing entities 2050
AIM EXPRESS-G entity (figure H.46) 2465
AIM short listing imported entity modifications 1744
mapping table 618, 677, 678, 680, 683, 1044, 1045, 1207, 1266, 1309, 1310
derived_unit_element
AIM expanded listing entities 2050
AIM EXPRESS-G entity (figure H.46) 2465
descriptive_measure
AIM expanded listing types 1987
AIM EXPRESS-G types (figure H.48) 2467
descriptive_representation_item
AIM expanded listing entities 2050
AIM EXPRESS-G entity (figure H.17) 2436
AIM short listing imported entity modifications 1744
mapping table .. 606, 609–611, 615, 617, 619, 620, 680, 694, 729–732, 735–737, 762, 771, 785, 840, 1041, 1042, 1204, 1329, 1345, 1346, 1350

Descriptive_specification
application assertion 460, 490
application object 144
ARM EXPRESS-G (figure G.8) 2375
mapping table 1345
Design_constraint
application assertion 400, 412, 444, 451, 460, 461, 465, 469, 476, 482, 508, 520, 531, 556, 564, 588, 593
application object 144
ARM EXPRESS-G (figure G.9) 2376
mapping table 1346
Design_constraint_association
application assertion 460, 461
application object 145
ARM EXPRESS-G (figure G.9) 2376
mapping table 1347
Design_constraint_relationship
application assertion 426, 461
application object 146
ARM EXPRESS-G (figure G.9) 2376
mapping table 1348
Design_constraint_version
application object 147
ARM EXPRESS-G (figure G.9) 2376
mapping table 1349
Design_discipline_item_definition
application assertion 412, 444, 451, 461, 462, 465, 476, 482, 490, 507, 508, 511, 520, 532, 541, 543, 546, 556, 571, 588
application object 147
ARM EXPRESS-G (figure G.5) 2372
mapping table 1120
Detailed_element
application assertion 462, 502, 516, 524
application object 148
ARM EXPRESS-G (figure G.35) 2402
mapping table 621, 647, 861, 868, 877, 892, 903, 920, 945, 1007
Detailed_geometric_model_element
application assertion 419–421, 504
application object 148
ARM EXPRESS-G (figure G.35) 2402
mapping table 862, 869, 877, 892, 903, 921, 945
Detailed_model_element
application assertion 579
application object 149
ARM EXPRESS-G (figure G.35) 2402
mapping table 862, 869, 877, 893, 903, 921, 945
diameter_dimension
AIM expanded listing entities 2050
AIM EXPRESS-G entity (figure H.106) 2525
mapping table 621, 624
Diameter_taper
application assertion 424, 462, 504, 538, 562

application object 149
ARM EXPRESS-G (figure G.29) 2396
mapping table 741
Digital_document
application assertion 458, 487
application object 150
ARM EXPRESS-G (figure G.10) 2377
mapping table 650
digital_document_reference
AIM expanded listing entities 2050
AIM EXPRESS-G entity (figure H.49) 2468
AIM short listing entities 1572
digital_document_reference_item
AIM expanded listing types 1987
AIM EXPRESS-G types (figure H.49) 2468
AIM short listing types 1506
Dimension
application assertion 462, 463
application object 151
ARM EXPRESS-G (figure G.33) 2400
mapping table 621
Dimension_callout
application assertion 462
application object 152
ARM EXPRESS-G (figure G.33) 2400
mapping table 624, 648
dimension_callout_component_relationship
AIM expanded listing entities 2051
AIM EXPRESS-G entity (figure H.107) 2526
dimension_callout_relationship
AIM expanded listing entities 2051
AIM EXPRESS-G entity (figure H.107) 2526
dimension_count
AIM expanded listing types 1987
AIM EXPRESS-G types (figure H.57) 2476
Dimension_curve
application assertion 462
application object 152
ARM EXPRESS-G (figure G.34) 2401
mapping table 624
dimension_curve
AIM expanded listing entities 2051
AIM EXPRESS-G entity (figure H.85) 2504
dimension_curve_directed_callout
AIM expanded listing entities 2052
AIM EXPRESS-G entity (figure H.106) 2525
dimension_curve_terminator
AIM expanded listing entities 2052
AIM EXPRESS-G entity (figure H.86) 2505
dimension_extent_usage
AIM expanded listing types 1987
AIM EXPRESS-G types (figure H.86) 2505

dimension_of
AIM expanded listing functions 2224
dimension_pair
AIM expanded listing entities 2052
AIM EXPRESS-G entity (figure H.107) 2526
mapping table 625
dimension_related_tolerance_zone_element
AIM expanded listing entities 2052
AIM EXPRESS-G entity (figure H.109) 2528
Dimension_sequence_pair
application assertion 463
application object 153
ARM EXPRESS-G (figure G.33) 2400
mapping table 625
dimension_text_associativity
AIM expanded listing entities 2053
AIM EXPRESS-G entity (figure H.17) 2436
Dimension_tolerance
application assertion 463, 464
application object 153
ARM EXPRESS-G (figure G.47) 2414
mapping table 1430
dimensional_characteristic
AIM expanded listing types 1987
AIM EXPRESS-G types (figure H.110) 2529
dimensional_characteristic_representation
AIM expanded listing entities 2053
AIM EXPRESS-G entity (figure H.20) 2439
mapping table 1430
dimensional_exponents
AIM expanded listing entities 2053
AIM EXPRESS-G entity (figure H.47) 2466
dimensional_location
AIM expanded listing entities 2053
AIM EXPRESS-G entity (figure H.12) 2431
AIM short listing imported entity modifications 1744
mapping table 1434, 1436
dimensional_location_with_path
AIM expanded listing entities 2053
AIM EXPRESS-G entity (figure H.12) 2431
mapping table 1427
dimensional_size
AIM expanded listing entities 2053
AIM EXPRESS-G entity (figure H.110) 2529
AIM short listing imported entity modifications 1745
mapping table 1435, 1440
dimensionality_is_two_or_three
AIM expanded listing rules 2261
AIM short listing rules 1796
mapping table 1163
dimensions_for_si_unit
AIM expanded listing functions 2224

directed_action
AIM expanded listing entities 2054
AIM EXPRESS-G entity (figure H.38) 2457
directed_angle
AIM expanded listing entities 2054
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1573
mapping table 680
directed_dimensional_location
AIM expanded listing entities 2054
AIM EXPRESS-G entity (figure H.12) 2431
AIM short listing entities 1574
mapping table 1436
Directed_taper
application assertion 424, 464, 504, 538, 561, 562
application object 155
ARM EXPRESS-G (figure G.29) 2396
mapping table 742
direction
AIM expanded listing entities 2054
AIM EXPRESS-G entity (figure H.59) 2478
mapping table 1036
direction_count_select
AIM expanded listing types 1987
AIM EXPRESS-G types (figure H.95) 2514
direction_shape_representation
AIM expanded listing entities 2055
AIM EXPRESS-G entity (figure H.16) 2435
AIM short listing entities 1574
mapping table 742, 783, 784, 791, 799, 804, 805
document
AIM expanded listing entities 2055
AIM EXPRESS-G entity (figure H.49) 2468
AIM short listing imported entity modifications 1745
mapping table 626, 627, 650, 655, 656, 667
document_reference
AIM expanded listing entities 2055
AIM EXPRESS-G entity (figure H.49) 2468
document_reference_item
AIM expanded listing types 1987
AIM EXPRESS-G types (figure H.51) 2470
AIM short listing types 1506
Document_relationship
application assertion 464
application object 155
ARM EXPRESS-G (figure G.10) 2377
mapping table 652
document_relationship
AIM expanded listing entities 2055
AIM EXPRESS-G entity (figure H.49) 2468
AIM short listing imported entity modifications 1745
mapping table 652, 653

document_role
AIM expanded listing entities . . . 2055
AIM EXPRESS-G entity (figure H.49) . . . 2468
AIM short listing imported entity modifications . . . 1745
mapping table . . . 658
Document_type
application assertion . . . 406, 464, 481
application object . . . 157
ARM EXPRESS-G (figure G.10) . . . 2377
mapping table . . . 654
document_type
AIM expanded listing entities . . . 2055
AIM EXPRESS-G entity (figure H.49) . . . 2468
AIM short listing imported entity modifications . . . 1746
mapping table . . . 654
document_usage_constraint
AIM expanded listing entities . . . 2055
AIM EXPRESS-G entity (figure H.49) . . . 2468
mapping table . . . 650–652, 656
dot_product
AIM expanded listing functions . . . 2225
draughting_annotation_occurrence
AIM expanded listing entities . . . 2056
AIM EXPRESS-G entity (figure H.85) . . . 2504
mapping table . . . 1005, 1012
Draughting_callout
application object . . . 158
ARM EXPRESS-G (figure G.34) . . . 2401
mapping table . . . 1007
draughting_callout
AIM expanded listing entities . . . 2058
AIM EXPRESS-G entity (figure H.106) . . . 2525
mapping table . . . 624, 648
draughting_callout_element
AIM expanded listing types . . . 1988
AIM EXPRESS-G types (figure H.106) . . . 2525
draughting_callout_relationship
AIM expanded listing entities . . . 2058
AIM EXPRESS-G entity (figure H.107) . . . 2526
draughting_elements
AIM expanded listing entities . . . 2058
AIM EXPRESS-G entity (figure H.106) . . . 2525
mapping table . . . 636, 1005, 1007, 1008, 1012, 1013
draughting_model
AIM expanded listing entities . . . 2058
AIM EXPRESS-G entity (figure H.16) . . . 2435
AIM short listing entities . . . 1575
mapping table . . . 991, 994, 1004
draughting_model_annotation_layers
AIM expanded listing rules . . . 2261
AIM short listing rules . . . 1797
mapping table . . . 993

draughting_pre_defined_colour
AIM expanded listing entities 2059
AIM EXPRESS-G entity (figure H.91) 2510
draughting_pre_defined_curve_font
AIM expanded listing entities 2059
AIM EXPRESS-G entity (figure H.55) 2474
draughting_pre_defined_text_font
AIM expanded listing entities 2059
AIM EXPRESS-G entity (figure H.55) 2474
AIM short listing entities 1577
draughting_specification_reference
AIM expanded listing entities 2059
AIM EXPRESS-G entity (figure H.49) 2468
AIM short listing entities 1577
draughting_subfigure_representation
AIM expanded listing entities 2059
AIM EXPRESS-G entity (figure H.86) 2505
AIM short listing imported entity modifications 1746
draughting_subfigure_representation_layers
AIM expanded listing rules 2261
AIM short listing rules 1798
mapping table 1169
draughting_symbol_representation
AIM expanded listing entities 2060
AIM EXPRESS-G entity (figure H.86) 2505
draughting_text_literal_with_delineation
AIM expanded listing entities 2060
AIM EXPRESS-G entity (figure H.89) 2508
draughting_title
AIM expanded listing entities 2061
AIM EXPRESS-G entity (figure H.107) 2526
AIM short listing imported entity modifications 1746
mapping table 627
draughting_titled_item
AIM expanded listing types 1988
AIM EXPRESS-G types (figure H.107) 2526
Drawing
application assertion 400, 412, 444, 451, 464, 465, 467, 469, 477, 532, 556, 564, 594
application object 158
ARM EXPRESS-G (figure G.32) 2399
mapping table 626
Drawing_assignment
application assertion 465–467
application object 160
ARM EXPRESS-G (figure G.32) 2399
mapping table 628
drawing_definition
AIM expanded listing entities 2061
AIM EXPRESS-G entity (figure H.107) 2526
mapping table 626, 654
drawing_revision
AIM expanded listing entities 2061

AIM EXPRESS-G entity (figure H.107) 2526
AIM short listing imported entity modifications 1746
mapping table 626
drawing_revision_sequence
AIM expanded listing entities 2061
AIM EXPRESS-G entity (figure H.107) 2526
mapping table 631
Drawing_sequence
application assertion 467
application object 160
ARM EXPRESS-G (figure G.32) 2399
mapping table 631
Drawing_sheet
application assertion 417, 439, 458, 467, 482, 587
application object 161
ARM EXPRESS-G (figure G.32) 2399
mapping table 631
drawing_sheet_annotation_layers
AIM expanded listing rules 2262
AIM short listing rules 1799
mapping table 1006
drawing_sheet_layout_usage
AIM expanded listing rules 2262
AIM short listing rules 1799
mapping table 998
drawing_sheet_revision
AIM expanded listing entities 2061
AIM EXPRESS-G entity (figure H.107) 2526
AIM short listing imported entity modifications 1746
mapping table 631
drawing_sheet_revision_usage
AIM expanded listing entities 2061
AIM EXPRESS-G entity (figure H.90) 2509
AIM short listing imported entity modifications 1747
mapping table 631
drawing_view_annotation_layers
AIM expanded listing rules 2262
AIM short listing rules 1800
mapping table 1013

edge
AIM expanded listing entities 2062
AIM EXPRESS-G entity (figure H.80) 2499
edge_based_wireframe_model
AIM expanded listing entities 2062
AIM EXPRESS-G entity (figure H.78) 2497
mapping table 932
edge_based_wireframe_shape_representation
AIM expanded listing entities 2062
AIM EXPRESS-G entity (figure H.20) 2439
mapping table 915
edge_curve

AIM expanded listing entities 2063
AIM EXPRESS-G entity (figure H.80) 2499
Edge_loop
application assertion 468, 487
application object 162
ARM EXPRESS-G (figure G.44) 2411
mapping table 877, 903, 921, 946
edge_loop
AIM expanded listing entities 2063
AIM EXPRESS-G entity (figure H.81) 2500
mapping table 921
edge_reversed
AIM expanded listing functions 2226
Edge_round
application object 162
ARM EXPRESS-G (figure G.29) 2396
mapping table 743
edge_round
AIM expanded listing entities 2063
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1578
mapping table 725, 743
Edge_transition
application assertion 468
application object 162
ARM EXPRESS-G (figure G.44) 2411
mapping table 877, 903, 921, 946
effective_context_assignment
AIM expanded listing entities 2064
AIM EXPRESS-G entity (figure H.36) 2455
AIM short listing imported entity modifications 1747
effective_context_assignment_subtype_exclusiveness
AIM expanded listing rules 2262
AIM short listing rules 1801
effective_context_item
AIM expanded listing types 1988
AIM EXPRESS-G types (figure H.37) 2456
AIM short listing types 1507
effective_item
AIM expanded listing types 1988
AIM EXPRESS-G types (figure H.37) 2456
AIM short listing types 1507
effective_item_assignment
AIM expanded listing entities 2064
AIM EXPRESS-G entity (figure H.36) 2455
AIM short listing imported entity modifications 1747
effective_item_assignment_subtype_exclusiveness
AIM expanded listing rules 2262
AIM short listing rules 1801
Effectivity
application assertion 468–474
application object 164

ARM EXPRESS-G (figure G.12) ... 2379
mapping table ... 1209
effectivity
AIM expanded listing entities ... 2064
AIM EXPRESS-G entity (figure H.36) ... 2455
AIM short listing imported entity modifications ... 1747
mapping table ... 1209
effectivity_assignment
AIM expanded listing entities ... 2065
AIM EXPRESS-G entity (figure H.36) ... 2455
Effectivity_relationship
application assertion ... 474
application object ... 166
ARM EXPRESS-G (figure G.12) ... 2379
mapping table ... 1226
effectivity_relationship
AIM expanded listing entities ... 2065
AIM EXPRESS-G entity (figure H.36) ... 2455
AIM short listing imported entity modifications ... 1748
mapping table ... 1226
effectivity_requires_effective_item_assignment
AIM expanded listing rules ... 2263
AIM short listing rules ... 1802
mapping table ... 1209
effectivity_role
AIM expanded listing entities ... 2065
AIM EXPRESS-G entity (figure H.36) ... 2455
AIM short listing imported entity modifications ... 1748
mapping table ... 1209
electric_current_measure
AIM expanded listing types ... 1988
AIM EXPRESS-G types (figure H.48) ... 2467
electric_current_measure_with_unit
AIM expanded listing entities ... 2065
AIM EXPRESS-G entity (figure H.46) ... 2465
electric_current_unit
AIM expanded listing entities ... 2065
AIM EXPRESS-G entity (figure H.47) ... 2466
Element_delivery
application assertion ... 474, 475
application object ... 167
ARM EXPRESS-G (figure G.16) ... 2383
mapping table ... 1296
element_delivery
AIM expanded listing entities ... 2065
AIM EXPRESS-G entity (figure H.38) ... 2457
AIM short listing entities ... 1580
mapping table ... 1296
element_delivery_assignment
AIM expanded listing entities ... 2066
AIM EXPRESS-G entity (figure H.41) ... 2460
AIM short listing entities ... 1581

element_delivery_item
AIM expanded listing types 1988
AIM EXPRESS-G types (figure H.41) 2460
AIM short listing types 1508
elementary_surface
AIM expanded listing entities 2066
AIM EXPRESS-G entity (figure H.70) 2489
Ellipse
application assertion 527
application object 168
ARM EXPRESS-G (figure G.43) 2410
mapping table 862, 869, 880, 905, 923, 948, 965
ellipse
AIM expanded listing entities 2067
AIM EXPRESS-G entity (figure H.62) 2481
mapping table 923
evaluated_degenerate_pcurve
AIM expanded listing entities 2067
AIM EXPRESS-G entity (figure H.58) 2477
event_assignment
AIM expanded listing entities 2067
AIM EXPRESS-G entity (figure H.41) 2460
AIM short listing entities 1582
event_based_effectivity
AIM expanded listing entities 2067
AIM EXPRESS-G entity (figure H.36) 2455
AIM short listing entities 1583
mapping table 1209, 1345–1348
event_context_assignment
AIM expanded listing entities 2067
AIM EXPRESS-G entity (figure H.41) 2460
AIM short listing entities 1584
event_context_item
AIM expanded listing types 1988
AIM EXPRESS-G types (figure H.44) 2463
AIM short listing types 1509
event_item
AIM expanded listing types 1989
AIM EXPRESS-G types (figure H.41) 2460
AIM short listing types 1510
Event_reference
application assertion 398, 470, 475–481, 549, 556
application object 168
ARM EXPRESS-G (figure G.12) 2379
mapping table 1227
event_reference
AIM expanded listing entities 2068
AIM EXPRESS-G entity (figure H.38) 2457
AIM short listing entities 1585
mapping table 1227
Exclusive_process_operation
application assertion 481

application object 169
ARM EXPRESS-G (figure G.19) 2386
mapping table 1384
exclusive_product_concept_feature_category
AIM expanded listing entities 2068
AIM EXPRESS-G entity (figure H.52) 2471
AIM short listing entities 1586
mapping table 1378
executed_action
AIM expanded listing entities 2068
AIM EXPRESS-G entity (figure H.38) 2457
expression_conversion_based_unit
AIM expanded listing entities 2068
AIM EXPRESS-G entity (figure H.47) 2466
expression_definition
AIM expanded listing entities 2068
AIM EXPRESS-G entity (figure H.47) 2466
expression_operand
AIM expanded listing types 1989
AIM EXPRESS-G types (figure H.47) 2466
extension
AIM expanded listing entities 2068
AIM EXPRESS-G entity (figure H.54) 2473
External_document
application assertion 400, 406, 412, 425, 431, 444, 451, 464, 470, 477, 481, 483, 492, 497, 532, 556, 565, 594
application object 169
ARM EXPRESS-G (figure G.10) 2377
mapping table 655
External_document_assignment
application assertion 481–486
application object 170
ARM EXPRESS-G (figure G.10) 2377
mapping table 658
External_document_version
application object 171
ARM EXPRESS-G (figure G.10) 2377
mapping table 667
external_identification_assignment
AIM expanded listing entities 2068
AIM EXPRESS-G entity (figure H.8) 2427
external_identification_item
AIM expanded listing types 1989
AIM EXPRESS-G types (figure H.8) 2427
AIM short listing types 1510
External_library_reference
application assertion 425, 431, 493
application object 171
ARM EXPRESS-G (figure G.10) 2377
mapping table 667
External_model
application assertion 487, 495, 497, 499, 525, 569, 582

application object 171
ARM EXPRESS-G (figure G.25) 2392
mapping table 668
external_source
AIM expanded listing entities 2069
AIM EXPRESS-G entity (figure H.50) 2469
mapping table 650, 651, 668
externally_defined_classification_attribute
AIM expanded listing entities 2069
AIM EXPRESS-G entity (figure H.50) 2469
AIM short listing entities 1586
mapping table 1329
externally_defined_classification_system
AIM expanded listing entities 2069
AIM EXPRESS-G entity (figure H.50) 2469
AIM short listing entities 1587
mapping table 1331
externally_defined_curve_font
AIM expanded listing entities 2069
AIM EXPRESS-G entity (figure H.50) 2469
AIM short listing imported entity modifications 1748
externally_defined_hatch_style
AIM expanded listing entities 2069
AIM EXPRESS-G entity (figure H.97) 2516
mapping table 1008–1010
externally_defined_item
AIM expanded listing entities 2069
AIM EXPRESS-G entity (figure H.50) 2469
mapping table 1331
externally_defined_product_category
AIM expanded listing entities 2070
AIM EXPRESS-G entity (figure H.2) 2421
AIM short listing entities 1588
mapping table 1333, 1335
externally_defined_representation
AIM expanded listing entities 2070
AIM EXPRESS-G entity (figure H.16) 2435
AIM short listing entities 1589
mapping table 668
externally_defined_symbol
AIM expanded listing entities 2070
AIM EXPRESS-G entity (figure H.50) 2469
AIM short listing imported entity modifications 1748
externally_defined_text_font
AIM expanded listing entities 2070
AIM EXPRESS-G entity (figure H.50) 2469
AIM short listing imported entity modifications 1749
externally_defined_thread
AIM expanded listing entities 2070
AIM EXPRESS-G entity (figure H.50) 2469
AIM short listing entities 1589
mapping table 694, 840, 841

externally_defined_tile_style
AIM expanded listing entities . . . 2070
AIM EXPRESS-G entity (figure H.97) . . . 2516
mapping table . . . 1010, 1011
extruded_face_solid
AIM expanded listing entities . . . 2071
AIM EXPRESS-G entity (figure H.74) . . . 2493
mapping table . . . 958, 960

Face
application assertion . . . 397, 432, 437, 487, 488, 571, 581, 585
application object . . . 172
ARM EXPRESS-G (figure G.44) . . . 2411
mapping table . . . 880, 906, 923, 948
face
AIM expanded listing entities . . . 2071
AIM EXPRESS-G entity (figure H.82) . . . 2501
face_based_surface_model
AIM expanded listing entities . . . 2071
AIM EXPRESS-G entity (figure H.78) . . . 2497
mapping table . . . 932, 933
Face_bound
application assertion . . . 487
application object . . . 173
ARM EXPRESS-G (figure G.44) . . . 2411
mapping table . . . 880, 906, 924, 948
face_bound
AIM expanded listing entities . . . 2071
AIM EXPRESS-G entity (figure H.82) . . . 2501
mapping table . . . 880, 906, 924, 948
face_bound_reversed
AIM expanded listing functions . . . 2226
face_outer_bound
AIM expanded listing entities . . . 2071
AIM EXPRESS-G entity (figure H.82) . . . 2501
mapping table . . . 881, 883, 884, 897, 906, 909, 910, 924, 927–929, 949, 958
face_reversed
AIM expanded listing functions . . . 2226
face_shape_representation
AIM expanded listing entities . . . 2071
AIM EXPRESS-G entity (figure H.16) . . . 2435
AIM short listing entities . . . 1590
face_surface
AIM expanded listing entities . . . 2071
AIM EXPRESS-G entity (figure H.82) . . . 2501
mapping table . . . 890, 923, 939, 948
Face_termination
application object . . . 173
ARM EXPRESS-G (figure G.38) . . . 2405
mapping table . . . 743
Face_transition
application assertion . . . 488

application object 174
ARM EXPRESS-G (figure G.44) 2411
mapping table 880, 906, 924, 949
Faceted_b_rep
application assertion 488
application object 175
ARM EXPRESS-G (figure G.45) 2412
mapping table 893, 951
Faceted_b_rep_model
application assertion 423, 441, 488, 574
application object 175
ARM EXPRESS-G (figure G.45) 2412
mapping table 893, 952
faceted_brep
AIM expanded listing entities 2072
AIM EXPRESS-G entity (figure H.75) 2494
mapping table 893, 951
faceted_brep_shape_representation
AIM expanded listing entities 2072
AIM EXPRESS-G entity (figure H.20) 2439
mapping table 893, 952
Faceted_closed_shell
application assertion 488
application object 175
ARM EXPRESS-G (figure G.45) 2412
mapping table 893, 952
Faceted_face
application assertion 488
application object 176
ARM EXPRESS-G (figure G.45) 2412
mapping table 893, 952
feature_component_definition
AIM expanded listing entities 2072
AIM EXPRESS-G entity (figure H.2) 2421
AIM short listing entities 1591
feature_component_relationship
AIM expanded listing entities 2073
AIM EXPRESS-G entity (figure H.12) 2431
AIM short listing entities 1592
mapping table 696
Feature_definition
application assertion 488, 489, 493, 504, 508, 520, 538, 554, 569
application object 176
ARM EXPRESS-G (figure G.27) 2394
mapping table 744
feature_definition
AIM expanded listing entities 2073
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1592
mapping table 744
feature_in_panel
AIM expanded listing entities 2073

AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1593
mapping table 749, 791
Feature_occurrence
application assertion 489
application object 177
ARM EXPRESS-G (figure G.27) 2394
mapping table 672, 745, 849
Feature_parameter
application assertion 408, 422, 425, 426, 428, 436, 437, 456, 457, 459, 462–464, 489–492, 494, 513, 517, 518, 526, 529, 538, 552, 553, 561, 563, 577, 578, 582, 586, 592
application object 177
ARM EXPRESS-G (figure G.27) 2394
mapping table 672
feature_pattern
AIM expanded listing entities 2075
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1597
mapping table 759
Feature_placement
application assertion 489, 538
application object 178
ARM EXPRESS-G (figure G.27) 2394
mapping table 849
Featured_shape
application assertion 489
application object 179
ARM EXPRESS-G (figure G.27) 2394
mapping table 854
featured_shape_representation
AIM expanded listing entities 2075
AIM EXPRESS-G entity (figure H.20) 2439
AIM short listing entities 1598
mapping table 854
Fill_area
application assertion 419–421, 489
application object 179
ARM EXPRESS-G (figure G.34) 2401
mapping table 1007
fill_area_style
AIM expanded listing entities 2076
AIM EXPRESS-G entity (figure H.97) 2516
AIM short listing imported entity modifications 1749
fill_area_style_colour
AIM expanded listing entities 2076
AIM EXPRESS-G entity (figure H.97) 2516
AIM short listing imported entity modifications 1749
mapping table 1009
fill_area_style_hatching
AIM expanded listing entities 2076
AIM EXPRESS-G entity (figure H.97) 2516
mapping table 1008

fill_area_style_tile_shape_select
AIM expanded listing types 1989
AIM EXPRESS-G types (figure H.98) 2517
fill_area_style_tile_symbol_constraint
AIM expanded listing rules 2263
AIM short listing rules 1803
mapping table 1010
fill_area_style_tile_symbol_with_style
AIM expanded listing entities 2076
AIM EXPRESS-G entity (figure H.98) 2517
AIM short listing imported entity modifications 1749
fill_area_style_tiles
AIM expanded listing entities 2076
AIM EXPRESS-G entity (figure H.98) 2517
mapping table 1010
fill_style_select
AIM expanded listing types 1989
AIM EXPRESS-G types (figure H.97) 2516
Fillet
application object 180
ARM EXPRESS-G (figure G.29) 2396
mapping table 745
fillet
AIM expanded listing entities 2076
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1598
mapping table 727, 745
Final_item_solution
application assertion 490
application object 180
ARM EXPRESS-G (figure G.8) 2375
mapping table 1350
first_proj_axis
AIM expanded listing functions 2227
Flat_bottom
application object 181
ARM EXPRESS-G (figure G.36) 2403
mapping table 745
Flat_slot_end_type
application assertion 490
application object 181
ARM EXPRESS-G (figure G.36) 2403
mapping table 746
Flat_with_chamfer_bottom
application assertion 490
application object 182
ARM EXPRESS-G (figure G.36) 2403
mapping table 747
Flat_with_radius_bottom
application assertion 490
application object 183
ARM EXPRESS-G (figure G.36) 2403

mapping table 748
Flatness_tolerance
application object 183
ARM EXPRESS-G (figure G.48) 2415
mapping table 1449
flatness_tolerance
AIM expanded listing entities 2077
AIM EXPRESS-G entity (figure H.109) 2528
mapping table 1449
font_select
AIM expanded listing types 1989
AIM EXPRESS-G types (figure H.89) 2508
Form_feature_in_panel
application assertion 490, 491
application object 183
ARM EXPRESS-G (figure G.39) 2406
mapping table 749
frame_associated_to_background
AIM expanded listing functions 2227
fully_constrained_pair
AIM expanded listing entities 2077
AIM EXPRESS-G entity (figure H.116) 2535
Function_component_association
application assertion 491
application object 185
ARM EXPRESS-G (figure G.7) 2374
mapping table 1351
functional_product_definition_requires_functional_product
AIM expanded listing rules 2263
AIM short listing rules 1804
mapping table 1367
functionally_defined_transformation
AIM expanded listing entities 2078
AIM EXPRESS-G entity (figure H.18) 2437
mapping table 1181, 1182

gbsf_check_curve
AIM expanded listing functions 2228
gbsf_check_point
AIM expanded listing functions 2230
gbsf_check_surface
AIM expanded listing functions 2230
gear_pair
AIM expanded listing entities 2078
AIM EXPRESS-G entity (figure H.118) 2537
gear_pair_range
AIM expanded listing entities 2078
AIM EXPRESS-G entity (figure H.120) 2539
gear_pair_value
AIM expanded listing entities 2078
AIM EXPRESS-G entity (figure H.112) 2531
General_boss

application assertion 492
application object 186
ARM EXPRESS-G (figure G.37) 2404
mapping table 754

General_bottom_condition
application assertion 492
application object 186
ARM EXPRESS-G (figure G.37) 2404
mapping table 755

General_classification
application assertion 492, 493
application object 187
ARM EXPRESS-G (figure G.1) 2368
mapping table 1333

General_classification_hierarchy
application assertion 493
application object 189
ARM EXPRESS-G (figure G.1) 2368
mapping table 1335

General_closed_profile
application assertion 492, 493
application object 189
ARM EXPRESS-G (figure G.30) 2397
mapping table 756

General_compound_feature
application assertion 493, 494
application object 190
ARM EXPRESS-G (figure G.27) 2394
mapping table 672

General_feature
application assertion 493, 494, 504, 508, 521, 570
application object 190
ARM EXPRESS-G (figure G.27) 2394
mapping table 674

General_feature_parameter
application assertion 494
application object 191
ARM EXPRESS-G (figure G.27) 2394
mapping table 676

General_item_definition_instance_relationship
application assertion 494
application object 192
ARM EXPRESS-G (figure G.13) 2380
mapping table 1191

General_item_definition_relationship
application assertion 495
application object 193
ARM EXPRESS-G (figure G.13) 2380
mapping table 1192

General_item_instance_relationship
application assertion 495
application object 194

ARM EXPRESS-G (figure G.14) ... 2381
mapping table ... 1193
General_item_property
application assertion ... 495, 496
application object ... 195
ARM EXPRESS-G (figure G.23) ... 2390
mapping table ... 1024
General_open_profile
application assertion ... 496
application object ... 195
ARM EXPRESS-G (figure G.30) ... 2397
mapping table ... 757
General_path
application assertion ... 496
application object ... 196
ARM EXPRESS-G (figure G.29) ... 2396
mapping table ... 758
General_pattern
application assertion ... 496
application object ... 196
ARM EXPRESS-G (figure G.40) ... 2407
mapping table ... 759
General_pocket_in_solid
application assertion ... 496
application object ... 197
ARM EXPRESS-G (figure G.37) ... 2404
mapping table ... 760
General_process_property
application assertion ... 497
application object ... 198
ARM EXPRESS-G (figure G.20) ... 2387
mapping table ... 1384
General_tolerance_table
application assertion ... 498, 583
application object ... 199
ARM EXPRESS-G (figure G.26) ... 2393
mapping table ... 1433
General_tolerances
application assertion ... 459, 497, 498
application object ... 199
ARM EXPRESS-G (figure G.26) ... 2393
mapping table ... 1433
General_top_condition
application assertion ... 498
application object ... 200
ARM EXPRESS-G (figure G.37) ... 2404
mapping table ... 761
geometric_alignment
AIM expanded listing entities ... 2078
AIM EXPRESS-G entity (figure H.54) ... 2473
geometric_curve_set
AIM expanded listing entities ... 2078

AIM EXPRESS-G entity (figure H.79) 2498
Geometric_dimension
application assertion 462, 463, 498
application object 201
ARM EXPRESS-G (figure G.47) 2414
mapping table 1434
geometric_intersection
AIM expanded listing entities 2079
AIM EXPRESS-G entity (figure H.54) 2473
Geometric_model
application assertion 397, 401, 406, 413, 438, 444, 451, 470, 477, 489, 495, 497–499, 502, 515, 532, 557, 565, 570, 579, 582, 594
application object 201
ARM EXPRESS-G (figure G.25) 2392
mapping table 862, 869, 883, 894, 909, 927, 952, 965
Geometric_model_relationship
application assertion 499, 524
application object 203
ARM EXPRESS-G (figure G.25) 2392
mapping table 863, 870, 884, 895, 910, 928, 954, 966
Geometric_model_relationship_with_transformation
application assertion 417, 499, 501, 505, 519, 563, 585
application object 204
ARM EXPRESS-G (figure G.25) 2392
mapping table 1164
Geometric_model_version
application object 204
ARM EXPRESS-G (figure G.25) 2392
mapping table 864, 871, 885, 896, 911, 929, 955, 967
geometric_representation_context
AIM expanded listing entities 2079
AIM EXPRESS-G entity (figure H.68) 2487
AIM short listing imported entity modifications 1749
mapping table 1163, 1164
geometric_representation_item
AIM expanded listing entities 2079
AIM EXPRESS-G entity (figure H.57) 2476
mapping table 862, 869, 877, 892, 903, 921, 945
geometric_set
AIM expanded listing entities 2079
AIM EXPRESS-G entity (figure H.79) 2498
Geometric_set_3d
application assertion 500, 502
application object 205
ARM EXPRESS-G (figure G.44) 2411
mapping table 967
geometric_set_select
AIM expanded listing types 1989
AIM EXPRESS-G types (figure H.79) 2498
Geometric_tolerance
application assertion 500, 501, 584
application object 205

ARM EXPRESS-G (figure G.48) ... 2415
mapping table ... 1449
geometric_tolerance
AIM expanded listing entities ... 2079
AIM EXPRESS-G entity (figure H.109) ... 2528
AIM short listing imported entity modifications ... 1750
mapping table ... 1449
Geometric_tolerance_precedence
application assertion ... 501
application object ... 207
ARM EXPRESS-G (figure G.48) ... 2415
mapping table ... 1452
geometric_tolerance_relationship
AIM expanded listing entities ... 2079
AIM EXPRESS-G entity (figure H.109) ... 2528
mapping table ... 1452
geometric_tolerance_subtype_exclusiveness
AIM expanded listing rules ... 2263
AIM short listing rules ... 1804
mapping table ... 1445, 1446, 1449, 1452, 1454, 1455, 1457, 1459, 1460, 1466
Geometric_tolerance_symbol
application object ... 207
ARM EXPRESS-G (figure G.33) ... 2400
mapping table ... 649
geometric_tolerance_with_datum_reference
AIM expanded listing entities ... 2080
AIM EXPRESS-G entity (figure H.110) ... 2529
geometric_tolerance_with_defined_unit
AIM expanded listing entities ... 2080
AIM EXPRESS-G entity (figure H.109) ... 2528
Geometrical_relationship
application assertion ... 501, 502
application object ... 207
ARM EXPRESS-G (figure G.13) ... 2380
mapping table ... 1194
geometrical_tolerance_callout
AIM expanded listing entities ... 2080
AIM EXPRESS-G entity (figure H.106) ... 2525
mapping table ... 649
geometrically_bounded_2d_wireframe_representation
AIM expanded listing entities ... 2080
AIM EXPRESS-G entity (figure H.20) ... 2439
mapping table ... 865
Geometrically_bounded_surface_model
application assertion ... 502
application object ... 209
ARM EXPRESS-G (figure G.44) ... 2411
mapping table ... 968
geometrically_bounded_surface_shape_representation
AIM expanded listing entities ... 2081
AIM EXPRESS-G entity (figure H.20) ... 2439
mapping table ... 968

geometrically_bounded_wireframe_shape_representati
AIM EXPRESS-G entity (figure H.20) 2439
geometrically_bounded_wireframe_shape_representation
AIM expanded listing entities 2081
mapping table 872
get_basis_surface
AIM expanded listing functions 2232
global_length_and_angle_units_2d_or_3d
AIM expanded listing rules 2264
AIM short listing rules 1806
mapping table 1163
global_uncertainty_assigned_context
AIM expanded listing entities 2082
mapping table 1161
global_unit_assigned_context
AIM expanded listing entities 2082
AIM EXPRESS-G entity (figure H.19) 2438
AIM short listing imported entity modifications 1750
mapping table 1163
Ground_representation
application assertion 502, 571
application object 209
ARM EXPRESS-G (figure G.21) 2388
mapping table 970
Group
application assertion 439, 502, 503, 578
application object 209
ARM EXPRESS-G (figure G.41) 2408
mapping table 1165
group
AIM expanded listing entities 2082
AIM EXPRESS-G entity (figure H.52) 2471
AIM short listing imported entity modifications 1750
mapping table 674, 675, 1165, 1331, 1378
group_assignment
AIM expanded listing entities 2083
AIM EXPRESS-G entity (figure H.52) 2471
AIM short listing imported entity modifications 1750
mapping table 1337
group_assignment_subtype_exclusiveness
AIM expanded listing rules 2264
AIM short listing rules 1806
mapping table 1272
group_item
AIM expanded listing types 1989
AIM EXPRESS-G types (figure H.52) 2471
AIM short listing types 1510
group_member_assignment
AIM expanded listing entities 2083
AIM EXPRESS-G entity (figure H.52) 2471
AIM short listing entities 1600
group_member_item

AIM expanded listing types 1990
AIM EXPRESS-G types (figure H.53) 2472
AIM short listing types 1510
Group_relationship
application assertion 503
application object 210
ARM EXPRESS-G (figure G.41) 2408
mapping table 1166
group_relationship
AIM expanded listing entities 2084
AIM EXPRESS-G entity (figure H.52) 2471
mapping table 1166, 1179
group_role
AIM expanded listing entities 2084
AIM EXPRESS-G entity (figure H.52) 2471
AIM short listing imported entity modifications 1751

Half_space_solid
application assertion 423
application object 211
ARM EXPRESS-G (figure G.46) 2413
mapping table 955
half_space_solid
AIM expanded listing entities 2084
AIM EXPRESS-G entity (figure H.78) 2497
Hardness
application assertion 503
application object 211
ARM EXPRESS-G (figure G.50) 2417
mapping table 609
Hatching
application assertion 409
application object 213
ARM EXPRESS-G (figure G.31) 2398
mapping table 1008
Hidden_element_handling
application assertion 503
application object 213
ARM EXPRESS-G (figure G.31) 2398
mapping table 1015
hidden_element_over_riding_styled_item
AIM expanded listing entities 2084
AIM EXPRESS-G entity (figure H.93) 2512
AIM short listing entities 1604
mapping table 1015, 1016
hiding_or_blanking_select
AIM expanded listing types 1990
AIM EXPRESS-G types (figure H.104) 2523
hole_bottom
AIM expanded listing entities 2085
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1604

mapping table 723, 724, 745–748, 762, 829, 841, 842
Hole_bottom_condition
application assertion 562
application object 214
ARM EXPRESS-G (figure G.36) 2403
mapping table 762
Hole_feature_in_panel
application assertion 504
application object 215
ARM EXPRESS-G (figure G.38) 2405
mapping table 763
hole_in_panel
AIM expanded listing entities 2087
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1608
mapping table 763, 779, 791, 792
hour_in_day
AIM expanded listing types 1990
AIM EXPRESS-G types (figure H.31) 2450
hybrid_3d_shape_representation
AIM expanded listing entities 2088
AIM EXPRESS-G entity (figure H.20) 2439
AIM short listing entities 1611
mapping table 872, 886, 897, 911, 929, 955
Hybrid_geometric_model_3d
application assertion 504
application object 216
ARM EXPRESS-G (figure G.35) 2402
mapping table 872, 886, 897, 911, 929, 955
Hyperbola
application assertion 527
application object 216
ARM EXPRESS-G (figure G.43) 2410
mapping table 865, 872, 886, 912, 929, 955, 968
hyperbola
AIM expanded listing entities 2088
AIM EXPRESS-G entity (figure H.62) 2481
mapping table 929

identification_assignment
AIM expanded listing entities 2088
AIM EXPRESS-G entity (figure H.8) 2427
AIM short listing imported entity modifications 1751
mapping table 603, 650, 651, 667, 672, 864, 871, 885, 896, 911, 929, 955, 967, 994, 1001, 1048, 1309, 1350, 1355, 1362, 1364, 1368, 1371, 1376, 1413
identification_assignment_subtype_exclusiveness
AIM expanded listing rules 2264
AIM short listing rules 1807
identification_item
AIM expanded listing types 1990
AIM EXPRESS-G types (figure H.8) 2427
AIM short listing types 1511

identification_role
AIM expanded listing entities 2088
AIM EXPRESS-G entity (figure H.8) 2427
AIM short listing imported entity modifications 1751
mapping table 1048
identifier
AIM expanded listing types 1990
Included_feature
application assertion 504, 505
application object 216
ARM EXPRESS-G (figure G.27) 2394
mapping table 676
Initial_configuration
application assertion 505
application object 216
ARM EXPRESS-G (figure G.22) 2389
mapping table 970
initial_state
AIM expanded listing entities 2088
AIM EXPRESS-G entity (figure H.111) 2530
mapping table 970
Instance_placement
application assertion 505, 506
application object 217
ARM EXPRESS-G (figure G.14) 2381
mapping table 1352
instanced_feature
AIM expanded listing entities 2088
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1611
mapping table 676, 677, 712, 717
Interpolated_configuration_sequence
application assertion 506, 516
application object 218
ARM EXPRESS-G (figure G.22) 2389
mapping table 971
interpolated_configuration_sequence
AIM expanded listing entities 2089
AIM EXPRESS-G entity (figure H.123) 2542
mapping table 971
interpolation_type
AIM expanded listing types 1990
AIM EXPRESS-G types (figure H.123) 2542
intersection_curve
AIM expanded listing entities 2089
AIM EXPRESS-G entity (figure H.66) 2485
invisibility
AIM expanded listing entities 2089
AIM EXPRESS-G entity (figure H.101) 2520
mapping table 991, 1005, 1012
invisibility_context
AIM expanded listing types 1990

AIM EXPRESS-G types (figure H.101) 2520
invisible_item
AIM expanded listing types 1990
AIM EXPRESS-G types (figure H.101) 2520
Item
application assertion 401, 408, 445, 451, 470, 477, 506, 513, 532, 557, 565, 594
application object 218
ARM EXPRESS-G (figure G.5) 2372
mapping table 1121
Item_classification
application assertion 483, 506
application object 219
ARM EXPRESS-G (figure G.1) 2368
mapping table 1122, 1335
Item_cost_property
application assertion 506
application object 220
ARM EXPRESS-G (figure G.23) 2390
mapping table 1026
item_defined_transformation
AIM expanded listing entities 2089
AIM EXPRESS-G entity (figure H.18) 2437
mapping table 1181, 1182
Item_definition_instance_relationship
application assertion 401, 413, 445, 452, 465, 470, 477, 483, 507, 509, 511, 521, 532, 557, 571, 588, 594
application object 220
ARM EXPRESS-G (figure G.13) 2380
mapping table 1198
Item_definition_relationship
application assertion 401, 413, 445, 452, 465, 470, 477, 483, 507, 509, 511, 521, 533, 557, 571, 588, 594
application object 221
ARM EXPRESS-G (figure G.13) 2380
mapping table 1199
item_in_context
AIM expanded listing functions 2233
Item_instance
application assertion 401, 406, 413, 445, 452, 466, 471, 478, 483, 507–509, 512, 521, 533, 537, 541, 557, 572, 575, 588, 595
application object 221
ARM EXPRESS-G (figure G.14) 2381
mapping table 1199
Item_instance_relationship
application assertion 401, 413, 445, 452, 466, 471, 478, 483, 508, 509, 512, 521, 533, 557, 572, 589, 595
application object 222
ARM EXPRESS-G (figure G.14) 2381
mapping table 1202
Item_property
application assertion 508–511
application object 223

ARM EXPRESS-G (figure G.23) 2390
mapping table 1026
Item_quality_property
application assertion 511
application object 223
ARM EXPRESS-G (figure G.23) 2390
mapping table 1030
Item_shape
application assertion 459, 489, 509, 511, 512, 521, 568–570, 580
application object 224
ARM EXPRESS-G (figure G.15) 2382
mapping table 1167
Item_solution
application assertion 402, 406, 413, 436, 445, 452, 460, 466, 471, 478, 484, 509, 512, 522, 533, 558, 565, 572, 575, 589, 595
application object 224
ARM EXPRESS-G (figure G.8) 2375
mapping table 1354
Item_solution_relationship
application assertion 512, 513
application object 225
ARM EXPRESS-G (figure G.8) 2375
mapping table 1355
Item_version
application assertion 402, 413, 440, 446, 452, 461, 471, 478, 513, 533, 537, 544, 548, 558, 565, 595
application object 226
ARM EXPRESS-G (figure G.5) 2372
mapping table 1123
Item_version_relationship
application assertion 426, 513
application object 227
ARM EXPRESS-G (figure G.5) 2372
mapping table 1298

joggle
AIM expanded listing entities 2089
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1612
mapping table 765, 791
Joggle_feature
application assertion 513, 514
application object 229
ARM EXPRESS-G (figure G.38) 2405
mapping table 765
Joggle_termination
application assertion 514
application object 230
ARM EXPRESS-G (figure G.38) 2405
mapping table 770
joggle_termination
AIM expanded listing entities 2091

AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1615
mapping table 743, 770, 843

kinematic_analysis_definition
AIM expanded listing types 1990
AIM EXPRESS-G types (figure H.124) 2543
kinematic_analysis_result
AIM expanded listing entities 2092
AIM EXPRESS-G entity (figure H.123) 2542
mapping table 975
Kinematic_configuration_definition
application assertion 514
application object 230
ARM EXPRESS-G (figure G.22) 2389
mapping table 971
Kinematic_configuration_interpolation
application assertion 506, 514
application object 231
ARM EXPRESS-G (figure G.22) 2389
mapping table 971
kinematic_frame_background
AIM expanded listing types 1991
AIM EXPRESS-G types (figure H.16) 2435
kinematic_frame_background_representation
AIM expanded listing entities 2092
AIM EXPRESS-G entity (figure H.16) 2435
kinematic_frame_background_representation_associat
AIM EXPRESS-G entity (figure H.18) 2437
kinematic_frame_background_representation_association
AIM expanded listing entities 2092
kinematic_frame_based_transformation
AIM expanded listing entities 2092
AIM EXPRESS-G entity (figure H.18) 2437
kinematic_ground_representation
AIM expanded listing entities 2092
AIM EXPRESS-G entity (figure H.16) 2435
mapping table 970
Kinematic_joint
application assertion 515, 516
application object 232
ARM EXPRESS-G (figure G.21) 2388
mapping table 972
kinematic_joint
AIM expanded listing entities 2093
AIM EXPRESS-G entity (figure H.111) 2530
Kinematic_link
application assertion 515
application object 233
ARM EXPRESS-G (figure G.21) 2388
mapping table 972
kinematic_link

AIM expanded listing entities 2093
AIM EXPRESS-G entity (figure H.111) 2530
Kinematic_link_representation
application assertion 515
application object 233
ARM EXPRESS-G (figure G.21) 2388
mapping table 973
kinematic_link_representation
AIM expanded listing entities 2093
AIM EXPRESS-G entity (figure H.111) 2530
mapping table 973
kinematic_link_representation_association
AIM expanded listing entities 2093
AIM EXPRESS-G entity (figure H.18) 2437
kinematic_link_representation_relation
AIM expanded listing entities 2094
AIM EXPRESS-G entity (figure H.111) 2530
Kinematic_mechanism
application assertion 505, 515, 516, 572
application object 233
ARM EXPRESS-G (figure G.21) 2388
mapping table 973
Kinematic_pair
application assertion 405, 515, 516, 528, 571
application object 234
ARM EXPRESS-G (figure G.21) 2388
mapping table 974
kinematic_pair
AIM expanded listing entities 2094
AIM EXPRESS-G entity (figure H.116) 2535
mapping table 974, 975
kinematic_property_definition
AIM expanded listing entities 2094
AIM EXPRESS-G entity (figure H.111) 2530
mapping table 977
kinematic_property_representation_relation
AIM expanded listing entities 2094
AIM EXPRESS-G entity (figure H.13) 2432
kinematic_result
AIM expanded listing types 1991
AIM EXPRESS-G types (figure H.123) 2542
Kinematic_structure
application assertion 516
application object 235
ARM EXPRESS-G (figure G.21) 2388
mapping table 975
kinematic_structure
AIM expanded listing entities 2094
AIM EXPRESS-G entity (figure H.111) 2530
mapping table 975
Kinematic_transformation_result
application assertion 516

application object 236
ARM EXPRESS-G (figure G.22) 2389
mapping table 975
knot_type
AIM expanded listing types 1991
AIM EXPRESS-G types (figure H.64) 2483
known_source
AIM expanded listing entities 2094
AIM EXPRESS-G entity (figure H.50) 2469
AIM short listing entities 1617
mapping table 667

label
AIM expanded listing types 1991
language_role
AIM expanded listing entities 2095
AIM EXPRESS-G entity (figure H.14) 2433
AIM short listing imported entity modifications 1751
mapping table 1159
Layer
application assertion 439, 516, 517, 578
application object 236
ARM EXPRESS-G (figure G.41) 2408
mapping table 1168
layered_item
AIM expanded listing types 1991
AIM EXPRESS-G types (figure H.90) 2509
Leader
application assertion 517
application object 237
ARM EXPRESS-G (figure G.34) 2401
mapping table 1008
leader_curve
AIM expanded listing entities 2095
AIM EXPRESS-G entity (figure H.85) 2504
mapping table 641
leader_directed_callout
AIM expanded listing entities 2095
AIM EXPRESS-G entity (figure H.106) 2525
leader_directed_dimension
AIM expanded listing entities 2095
AIM EXPRESS-G entity (figure H.106) 2525
mapping table 621
leader_terminator
AIM expanded listing entities 2095
AIM EXPRESS-G entity (figure H.86) 2505
leap_year
AIM expanded listing functions 2233
length_measure
AIM expanded listing types 1991
AIM EXPRESS-G types (figure H.48) 2467
length_measure_with_unit

AIM expanded listing entities 2095
AIM EXPRESS-G entity (figure H.46) 2465
mapping table 1434
length_unit
AIM expanded listing entities 2095
AIM EXPRESS-G entity (figure H.47) 2466
Light_source
application assertion 517
application object 238
ARM EXPRESS-G (figure G.31) 2398
mapping table 1016
light_source
AIM expanded listing entities 2096
AIM EXPRESS-G entity (figure H.91) 2510
mapping table 1016
light_source_ambient
AIM expanded listing entities 2096
AIM EXPRESS-G entity (figure H.91) 2510
mapping table 1017
light_source_directional
AIM expanded listing entities 2096
AIM EXPRESS-G entity (figure H.91) 2510
mapping table 1017
light_source_positional
AIM expanded listing entities 2096
AIM EXPRESS-G entity (figure H.91) 2510
mapping table 1017
light_source_spot
AIM expanded listing entities 2096
AIM EXPRESS-G entity (figure H.91) 2510
mapping table 1017
limit_condition
AIM expanded listing types 1991
AIM EXPRESS-G types (figure H.109) 2528
mapping table 1453
Limits_and_fits
application assertion 463
application object 238
ARM EXPRESS-G (figure G.47) 2414
mapping table 1435
limits_and_fits
AIM expanded listing entities 2096
AIM EXPRESS-G entity (figure H.110) 2529
mapping table 1435, 1436
Line
application assertion 528
application object 239
ARM EXPRESS-G (figure G.43) 2410
mapping table 865, 872, 886, 912, 929, 955, 968
line
AIM expanded listing entities 2097
AIM EXPRESS-G entity (figure H.62) 2481

mapping table 929
Linear_bead_end_type
application object 239
ARM EXPRESS-G (figure G.38) 2405
mapping table 770
linear_dimension
AIM expanded listing entities 2097
AIM EXPRESS-G entity (figure H.106) 2525
mapping table 621
Linear_distance_dimension
application object 240
ARM EXPRESS-G (figure G.47) 2414
mapping table 1436
Linear_path
application assertion 425, 514, 517, 539, 563
application object 240
ARM EXPRESS-G (figure G.29) 2396
mapping table 772
Linear_profile_tolerance
application assertion 517
application object 241
ARM EXPRESS-G (figure G.48) 2415
mapping table 1452
linear_profile_tolerance
AIM expanded listing entities 2097
AIM EXPRESS-G entity (figure H.109) 2528
mapping table 1452
list_face_loops
AIM expanded listing functions 2233
list_of_reversible_topology_item
AIM expanded listing types 1991
AIM EXPRESS-G types (figure H.84) 2503
list_of_topology_reversed
AIM expanded listing functions 2234
list_representation_item
AIM expanded listing types 1991
AIM EXPRESS-G types (figure H.17) 2436
list_to_array
AIM expanded listing functions 2234
list_to_set
AIM expanded listing functions 2234
local_time
AIM expanded listing entities 2097
AIM EXPRESS-G entity (figure H.31) 2450
mapping table 1104
Location_dimension
application assertion 517, 518
application object 242
ARM EXPRESS-G (figure G.47) 2414
mapping table 1436
location_shape_representation
AIM expanded listing entities 2097

AIM EXPRESS-G entity (figure H.16) 2435
AIM short listing entities 1618
mapping table 783, 784
locator
AIM expanded listing entities 2097
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1618
mapping table 773, 791
Locator_feature
application assertion 518
application object 242
ARM EXPRESS-G (figure G.39) 2406
mapping table 773
loop
AIM expanded listing entities 2098
AIM EXPRESS-G entity (figure H.81) 2500
Lot_configuration
application object 243
ARM EXPRESS-G (figure G.8) 2375
mapping table 1356
lot_effectivity
AIM expanded listing entities 2098
AIM EXPRESS-G entity (figure H.36) 2455
mapping table 1356, 1357
luminous_intensity_measure
AIM expanded listing types 1991
AIM EXPRESS-G types (figure H.48) 2467
luminous_intensity_measure_with_unit
AIM expanded listing entities 2098
AIM EXPRESS-G entity (figure H.46) 2465
luminous_intensity_unit
AIM expanded listing entities 2098
AIM EXPRESS-G entity (figure H.47) 2466

make_array_of_array
AIM expanded listing functions 2234
Make_from_relationship
application assertion 519
application object 243
ARM EXPRESS-G (figure G.20) 2387
mapping table 1385
make_from_usage_option
AIM expanded listing entities 2099
AIM EXPRESS-G entity (figure H.6) 2425
AIM short listing imported entity modifications 1751
mapping table 1385
manifold_solid_brep
AIM expanded listing entities 2099
AIM EXPRESS-G entity (figure H.75) 2494
manifold_surface_shape_representation
AIM expanded listing entities 2099
AIM EXPRESS-G entity (figure H.20) 2439

mapping table 888
mapped_item
AIM expanded listing entities 2102
AIM EXPRESS-G entity (figure H.17) 2436
AIM short listing imported entity modifications 1752
mapping table 849, 1180–1182
marker_select
AIM expanded listing types 1991
AIM EXPRESS-G types (figure H.94) 2513
marker_type
AIM expanded listing types 1991
AIM EXPRESS-G types (figure H.94) 2513
Mass
application assertion 519
application object 244
ARM EXPRESS-G (figure G.23) 2390
mapping table 1030
mass_measure
AIM expanded listing types 1992
AIM EXPRESS-G types (figure H.48) 2467
mass_measure_with_unit
AIM expanded listing entities 2102
AIM EXPRESS-G entity (figure H.46) 2465
mass_unit
AIM expanded listing entities 2102
AIM EXPRESS-G entity (figure H.47) 2466
Mated_item_association
application assertion 519, 541
application object 244
ARM EXPRESS-G (figure G.20) 2387
mapping table 1386
Mated_item_relationship
application assertion 519, 520
application object 245
ARM EXPRESS-G (figure G.20) 2387
mapping table 1387
mated_item_relationship
AIM expanded listing entities 2102
AIM EXPRESS-G entity (figure H.5) 2424
AIM short listing entities 1620
mapping table 1387
Material
application assertion 402, 414, 432, 446, 453, 471, 478, 484, 520–523, 533, 558, 565, 595
application object 246
ARM EXPRESS-G (figure G.24) 2391
mapping table 1031
Material_condition
application assertion 455, 501
application object 247
ARM EXPRESS-G (figure G.49) 2416
mapping table 1453
material_designation

AIM expanded listing entities 2103
AIM EXPRESS-G entity (figure H.56) 2475
mapping table 1031
material_designation_characterization
AIM expanded listing entities 2103
AIM EXPRESS-G entity (figure H.56) 2475
Material_property
application assertion 522, 523
application object 247
ARM EXPRESS-G (figure G.24) 2391
mapping table 1035
material_property
AIM expanded listing entities 2103
AIM EXPRESS-G entity (figure H.10) 2429
AIM short listing imported entity modifications 1752
mapping table 1035
Material_property_representation
application assertion 523, 524
application object 248
ARM EXPRESS-G (figure G.24) 2391
mapping table 1035
material_property_representation
AIM expanded listing entities 2103
AIM EXPRESS-G entity (figure H.56) 2475
mapping table 1035
Material_side_specification
application assertion 524, 583
application object 248
ARM EXPRESS-G (figure G.24) 2391
mapping table 1036
Mating_definition
application assertion 519
application object 249
ARM EXPRESS-G (figure G.20) 2387
mapping table 1393
measure_qualification
AIM expanded listing entities 2103
measure_representation_item
AIM expanded listing entities 2104
AIM EXPRESS-G entity (figure H.46) 2465
AIM short listing imported entity modifications 1752
mapping table.. 605, 616, 670, 672, 676, 683, 700, 701, 704, 771, 776, 799–802, 804, 805, 824, 838, 839, 1041, 1381, 1419, 1434, 1435, 1440, 1443, 1444, 1462–1464
measure_value
AIM expanded listing types 1992
AIM EXPRESS-G types (figure H.48) 2467
mapping table 618, 678, 1045, 1046, 1207, 1266, 1309
measure_with_unit
AIM expanded listing entities 2104
AIM EXPRESS-G entity (figure H.46) 2465
AIM short listing imported entity modifications 1752
mapping table.. 1044, 1266, 1357, 1397–1399, 1444, 1445, 1449, 1450, 1454, 1455, 1459, 1466

mechanical_design_geometric_presentation_area
AIM expanded listing entities 2104
AIM EXPRESS-G entity (figure H.90) 2509
mapping table 1001
mechanical_design_geometric_presentation_represent
AIM EXPRESS-G entity (figure H.16) 2435
mechanical_design_geometric_presentation_representation
AIM expanded listing entities 2104
AIM short listing imported entity modifications 1753
mapping table 991, 992, 994
mechanism
AIM expanded listing entities 2108
AIM EXPRESS-G entity (figure H.111) 2530
mapping table 973
mechanism_base_placement
AIM expanded listing entities 2108
AIM EXPRESS-G entity (figure H.105) 2524
minute_in_hour
AIM expanded listing types 1992
AIM EXPRESS-G types (figure H.31) 2450
mixed_loop_type_set
AIM expanded listing functions 2235
Model_change
application assertion 524, 525
application object 250
ARM EXPRESS-G (figure G.18) 2385
mapping table 1169
model_change
AIM expanded listing entities 2108
AIM EXPRESS-G entity (figure H.10) 2429
AIM short listing entities 1621
mapping table 1169
Model_image
application assertion 439, 503, 517, 525
application object 252
ARM EXPRESS-G (figure G.31) 2398
mapping table 987
modified_geometric_tolerance
AIM expanded listing entities 2109
AIM EXPRESS-G entity (figure H.109) 2528
modified_pattern
AIM expanded listing entities 2109
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1623
Moments_of_inertia
application assertion 525, 526
application object 253
ARM EXPRESS-G (figure G.24) 2391
mapping table 1037
month_in_year_number
AIM expanded listing types 1992
AIM EXPRESS-G types (figure H.31) 2450

motion_parameter_measure
AIM expanded listing types 1992
AIM EXPRESS-G types (figure H.123) 2542
msb_shells
AIM expanded listing functions 2235
multi_language_attribute_assignment
AIM expanded listing entities 2113
AIM EXPRESS-G entity (figure H.14) 2433
mapping table 1159
multi_language_attribute_item
AIM expanded listing types 1992
AIM EXPRESS-G types (figure H.15) 2434
AIM short listing types 1511
Multi_language_string
application assertion 398, 404–406, 408, 410, 414, 418, 425, 426, 429–433, 438, 442, 456, 457, 459–464, 467, 471, 474, 481, 487, 489, 493–499, 501–503, 506, 507, 511–513, 517, 519, 523–526, 537, 540, 542–551, 553, 558, 563, 566, 568, 569, 573, 575–577, 579, 581, 583, 585–587, 590–592, 595
application object 254
ARM EXPRESS-G (figure G.4) 2371
mapping table 1123

named_unit
AIM expanded listing entities 2113
AIM EXPRESS-G entity (figure H.47) 2466
AIM short listing imported entity modifications 1753
mapping table 618, 677, 1044, 1207, 1266, 1309
next_assembly_usage_occurrence
AIM expanded listing entities 2114
AIM EXPRESS-G entity (figure H.6) 2425
AIM short listing imported entity modifications 1753
mapping table 1203
Next_higher_assembly
application assertion 577
application object 254
ARM EXPRESS-G (figure G.13) 2380
mapping table 1203
ngon_closed_profile
AIM expanded listing entities 2114
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1631
mapping table 708, 709, 711, 776, 792
Ngon_profile
application assertion 526
application object 254
ARM EXPRESS-G (figure G.30) 2397
mapping table 776
nmsf_curve_check
AIM expanded listing functions 2235
nmsf_surface_check
AIM expanded listing functions 2237
non_manifold_surface_shape_representation
AIM expanded listing entities 2115

AIM EXPRESS-G entity (figure H.20) 2439
mapping table 915
normalise
AIM expanded listing functions 2237
null_style
AIM expanded listing types 1993
AIM EXPRESS-G types (figure H.93) 2512
numeric_measure
AIM expanded listing types 1993
AIM EXPRESS-G types (figure H.48) 2467

occurrence_usage_definition
AIM expanded listing entities 2117
AIM EXPRESS-G entity (figure H.3) 2422
AIM short listing entities 1633
Offset_curve
application object 256
ARM EXPRESS-G (figure G.43) 2410
mapping table 865, 872, 968
offset_curve_2d
AIM expanded listing entities 2117
AIM EXPRESS-G entity (figure H.67) 2486
offset_curve_3d
AIM expanded listing entities 2118
AIM EXPRESS-G entity (figure H.67) 2486
Offset_surface
application object 256
ARM EXPRESS-G (figure G.44) 2411
mapping table 886, 930, 968
offset_surface
AIM expanded listing entities 2118
AIM EXPRESS-G entity (figure H.71) 2490
mapping table 930
one_direction_repeat_factor
AIM expanded listing entities 2118
AIM EXPRESS-G entity (figure H.98) 2517
Open_bead_end_type
application object 256
ARM EXPRESS-G (figure G.38) 2405
mapping table 778
open_path_profile
AIM expanded listing entities 2118
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1634
mapping table 757, 778, 792
Open_profile
application assertion 422, 561, 574
application object 256
ARM EXPRESS-G (figure G.30) 2397
mapping table 778
open_shell
AIM expanded listing entities 2119

AIM EXPRESS-G entity (figure H.83) 2502
Open_slot_end_type
application object 256
ARM EXPRESS-G (figure G.36) 2403
mapping table 778
ordinate_dimension
AIM expanded listing entities 2119
AIM EXPRESS-G entity (figure H.106) 2525
mapping table 621
Organization
application assertion 398, 406, 449, 472, 475, 526, 527, 529, 534, 541, 544
application object 256
ARM EXPRESS-G (figure G.2) 2369
mapping table 1124
organization
AIM expanded listing entities 2119
AIM EXPRESS-G entity (figure H.24) 2443
mapping table 1102, 1124, 1301, 1382
organization_assignment
AIM expanded listing entities 2119
AIM EXPRESS-G entity (figure H.24) 2443
AIM short listing imported entity modifications 1753
organization_assignment_subtype_exclusiveness
AIM expanded listing rules 2265
AIM short listing rules 1808
Organization_in_contract
application assertion 526
application object 258
ARM EXPRESS-G (figure G.17) 2384
mapping table 1298
organization_item
AIM expanded listing types 1993
AIM EXPRESS-G types (figure H.30) 2449
AIM short listing types 1513
Organization_relationship
application assertion 527
application object 259
ARM EXPRESS-G (figure G.2) 2369
mapping table 1300
organization_relationship
AIM expanded listing entities 2119
AIM EXPRESS-G entity (figure H.24) 2443
AIM short listing imported entity modifications 1754
mapping table 1300, 1308
organization_role
AIM expanded listing entities 2119
AIM EXPRESS-G entity (figure H.24) 2443
AIM short listing imported entity modifications 1754
mapping table 1076, 1125, 1299
organizational_address
AIM expanded listing entities 2119
AIM EXPRESS-G entity (figure H.24) 2443

mapping table 669, 1124
oriented_closed_shell
AIM expanded listing entities 2119
AIM EXPRESS-G entity (figure H.83) 2502
mapping table 898, 939
oriented_edge
AIM expanded listing entities 2120
AIM EXPRESS-G entity (figure H.80) 2499
mapping table 886, 912, 916, 930, 945, 956
Oriented_edge_curve
application assertion 397, 437, 468, 488, 527, 528, 585
application object 259
ARM EXPRESS-G (figure G.44) 2411
mapping table 886, 912, 930, 956
oriented_face
AIM expanded listing entities 2120
AIM EXPRESS-G entity (figure H.82) 2501
mapping table 939
oriented_open_shell
AIM expanded listing entities 2120
AIM EXPRESS-G entity (figure H.83) 2502
oriented_path
AIM expanded listing entities 2120
AIM EXPRESS-G entity (figure H.81) 2500
orthogonal_complement
AIM expanded listing functions 2238
outer_boundary_curve
AIM expanded listing entities 2121
AIM EXPRESS-G entity (figure H.65) 2484
over_riding_styled_item
AIM expanded listing entities 2121
AIM EXPRESS-G entity (figure H.93) 2512

package_product_concept_feature
AIM expanded listing entities 2121
AIM EXPRESS-G entity (figure H.7) 2426
AIM short listing entities 1635
mapping table 1376
pair_actuator
AIM expanded listing entities 2121
AIM EXPRESS-G entity (figure H.116) 2535
AIM short listing imported entity modifications 1754
Pair_value
application assertion 505, 514, 528
application object 260
ARM EXPRESS-G (figure G.22) 2389
mapping table 976
pair_value
AIM expanded listing entities 2121
AIM EXPRESS-G entity (figure H.112) 2531
mapping table 976
Panel_hole_with_barring

application assertion 528
application object 261
ARM EXPRESS-G (figure G.38) 2405
mapping table 779
Parabola
application assertion 528
application object 261
ARM EXPRESS-G (figure G.43) 2410
mapping table 865, 872, 888, 914, 931, 957, 968
parabola
AIM expanded listing entities 2121
AIM EXPRESS-G entity (figure H.62) 2481
mapping table 931
parallel_offset
AIM expanded listing entities 2121
AIM EXPRESS-G entity (figure H.54) 2473
Parallelism_tolerance
application assertion 528, 529
application object 261
ARM EXPRESS-G (figure G.48) 2415
mapping table 1454
parallelism_tolerance
AIM expanded listing entities 2122
AIM EXPRESS-G entity (figure H.110) 2529
mapping table 1454
parameter_value
AIM expanded listing types 1993
AIM EXPRESS-G types (figure H.48) 2467
mapping table 977
parametric_representation_context
AIM expanded listing entities 2122
AIM EXPRESS-G entity (figure H.19) 2438
Partial_circular_path
application assertion 529
application object 262
ARM EXPRESS-G (figure G.29) 2396
mapping table 779
Partial_circular_profile
application assertion 529
application object 263
ARM EXPRESS-G (figure G.30) 2397
mapping table 780
partial_circular_profile
AIM expanded listing entities 2122
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1636
mapping table 778–780, 792
Path
application assertion 422, 529, 561, 574
application object 264
ARM EXPRESS-G (figure G.29) 2396
mapping table 782

path
AIM expanded listing entities 2122
AIM EXPRESS-G entity (figure H.81) 2500
Path_approximation
application assertion 514
application object 264
ARM EXPRESS-G (figure G.22) 2389
mapping table 977
path_feature_component
AIM expanded listing entities 2123
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1637
mapping table 702, 711, 758, 772, 779, 782
path_head_to_tail
AIM expanded listing functions 2239
path_reversed
AIM expanded listing functions 2239
path_shape_representation
AIM expanded listing entities 2124
AIM EXPRESS-G entity (figure H.16) 2435
AIM short listing entities 1641
pcurve
AIM expanded listing entities 2124
AIM EXPRESS-G entity (figure H.67) 2486
mapping table 876, 902, 920, 945, 946, 964
pcurve_or_surface
AIM expanded listing types 1994
AIM EXPRESS-G types (figure H.66) 2485
perpendicular_to
AIM expanded listing entities 2125
AIM EXPRESS-G entity (figure H.54) 2473
Perpendicularity_tolerance
application assertion 529
application object 264
ARM EXPRESS-G (figure G.48) 2415
mapping table 1455
perpendicularity_tolerance
AIM expanded listing entities 2125
AIM EXPRESS-G entity (figure H.110) 2529
mapping table 1455
Person
application assertion 449, 529, 534
application object 265
ARM EXPRESS-G (figure G.2) 2369
mapping table 1124
person
AIM expanded listing entities 2125
AIM EXPRESS-G entity (figure H.23) 2442
AIM short listing imported entity modifications 1754
mapping table 1102, 1104, 1124
person_and_organization
AIM expanded listing entities 2125

AIM EXPRESS-G entity (figure H.23) 2442
mapping table 1102, 1125
person_and_organization_assignment
AIM expanded listing entities 2125
AIM EXPRESS-G entity (figure H.23) 2442
person_and_organization_item
AIM expanded listing types 1994
AIM EXPRESS-G types (figure H.25) 2444
AIM short listing types 1514
person_and_organization_role
AIM expanded listing entities 2125
AIM EXPRESS-G entity (figure H.23) 2442
AIM short listing imported entity modifications 1754
mapping table 1076, 1125
person_assignment
AIM expanded listing entities 2125
AIM EXPRESS-G entity (figure H.23) 2442
Person_in_organization
application assertion 449, 529, 534
application object 266
ARM EXPRESS-G (figure G.2) 2369
mapping table 1125
person_item
AIM expanded listing types 1994
AIM EXPRESS-G types (figure H.29) 2448
AIM short listing types 1515
Person_or_organization_assignment
application assertion 530–537
application object 267
ARM EXPRESS-G (figure G.2) 2369
mapping table 1125
person_organization_select
AIM expanded listing types 1994
AIM EXPRESS-G types (figure H.23) 2442
person_requires_person_and_organization
AIM expanded listing rules 2265
AIM short listing rules 1809
mapping table 1124
person_role
AIM expanded listing entities 2126
AIM EXPRESS-G entity (figure H.23) 2442
AIM short listing imported entity modifications 1755
mapping table 1076, 1125
personal_address
AIM expanded listing entities 2126
AIM EXPRESS-G entity (figure H.24) 2443
mapping table 1124
Physical_instance
application assertion 402, 407, 414, 446, 453, 478, 484, 510, 522, 534, 537, 558, 566, 596
application object 268
ARM EXPRESS-G (figure G.9) 2376
mapping table 1357

Physical_instance_usage
application assertion 402, 414, 446, 453, 478, 484, 534, 537, 558, 566, 596
application object 269
ARM EXPRESS-G (figure G.9) 2376
mapping table 1358
Physical_model
application assertion 439
application object 269
ARM EXPRESS-G (figure G.10) 2377
mapping table 669
physically_modelled_product_definition
AIM expanded listing entities 2126
AIM EXPRESS-G entity (figure H.4) 2423
AIM short listing entities 1641
mapping table 655, 656, 658, 669
placed_datum_target_feature
AIM expanded listing entities 2126
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1643
mapping table 1456, 1461–1463
Placed_feature
application assertion 489, 538
application object 270
ARM EXPRESS-G (figure G.27) 2394
mapping table 855
placed_feature
AIM expanded listing entities 2128
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1646
mapping table 712, 717, 723, 855, 856
Placed_target
application assertion 538
application object 271
ARM EXPRESS-G (figure G.49) 2416
mapping table 1456
placement
AIM expanded listing entities 2128
AIM EXPRESS-G entity (figure H.59) 2478
mapping table 1163, 1436
Planar_bottom_condition
application assertion 538
application object 271
ARM EXPRESS-G (figure G.37) 2404
mapping table 782
planar_box
AIM expanded listing entities 2128
AIM EXPRESS-G entity (figure H.87) 2506
mapping table 989–991
planar_curve_pair
AIM expanded listing entities 2129
AIM EXPRESS-G entity (figure H.117) 2536
planar_curve_pair_range

AIM expanded listing entities 2129
AIM EXPRESS-G entity (figure H.119) 2538
planar_extent
AIM expanded listing entities 2129
AIM EXPRESS-G entity (figure H.87) 2506
planar_pair
AIM expanded listing entities 2129
AIM EXPRESS-G entity (figure H.116) 2535
planar_pair_range
AIM expanded listing entities 2129
AIM EXPRESS-G entity (figure H.120) 2539
planar_pair_value
AIM expanded listing entities 2130
AIM EXPRESS-G entity (figure H.113) 2532
Planar_top_condition
application object 272
ARM EXPRESS-G (figure G.37) 2404
mapping table 784
Plane
application assertion 488
application object 272
ARM EXPRESS-G (figure G.44) 2411
mapping table 888, 897, 914, 932, 957, 968
plane
AIM expanded listing entities 2130
AIM EXPRESS-G entity (figure H.70) 2489
mapping table 932
plane_angle_for_pair_in_radian
AIM expanded listing functions 2239
plane_angle_measure
AIM expanded listing types 1994
AIM EXPRESS-G types (figure H.48) 2467
plane_angle_measure_with_unit
AIM expanded listing entities 2130
AIM EXPRESS-G entity (figure H.46) 2465
mapping table 1434, 1435
plane_angle_unit
AIM expanded listing entities 2130
AIM EXPRESS-G entity (figure H.47) 2466
Plus_minus_tolerance
application assertion 463
application object 273
ARM EXPRESS-G (figure G.47) 2414
mapping table 1439
plus_minus_tolerance
AIM expanded listing entities 2130
AIM EXPRESS-G entity (figure H.110) 2529
Plus_minus_toleranced_datum
application assertion 584
application object 273
ARM EXPRESS-G (figure G.26) 2393
mapping table 1440

pocket
AIM expanded listing entities 2131
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1647
mapping table 760, 786, 792, 795, 803
pocket_bottom
AIM expanded listing entities 2132
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1651
mapping table 755, 782, 783, 785, 841
Pocket_bottom_condition
application assertion 539
application object 273
ARM EXPRESS-G (figure G.37) 2404
mapping table 785
Pocket_feature_in_solid
application assertion 538, 539
application object 274
ARM EXPRESS-G (figure G.37) 2404
mapping table 786
Point
application assertion 437, 441, 442, 500, 586
application object 275
ARM EXPRESS-G (figure G.43) 2410
mapping table 865, 872, 888, 897, 914, 932, 957, 968
point
AIM expanded listing entities 2134
AIM EXPRESS-G entity (figure H.58) 2477
mapping table 1021
Point_appearance
application assertion 409
application object 275
ARM EXPRESS-G (figure G.31) 2398
mapping table 990
Point_direction
application assertion 539
application object 275
ARM EXPRESS-G (figure G.42) 2409
mapping table 982
Point_direction_model
application assertion 539
application object 277
ARM EXPRESS-G (figure G.42) 2409
mapping table 982
Point_direction_sequence
application assertion 539
application object 277
ARM EXPRESS-G (figure G.42) 2409
mapping table 982
Point_on_curve
application object 278
ARM EXPRESS-G (figure G.43) 2410

mapping table 865, 872, 888, 932, 968
point_on_curve
AIM expanded listing entities 2134
AIM EXPRESS-G entity (figure H.58) 2477
mapping table 859, 860, 865–867, 872–875, 888, 916, 932, 962, 963, 968
point_on_planar_curve_pair
AIM expanded listing entities 2134
AIM EXPRESS-G entity (figure H.115) 2534
point_on_planar_curve_pair_range
AIM expanded listing entities 2134
AIM EXPRESS-G entity (figure H.119) 2538
point_on_planar_curve_pair_value
AIM expanded listing entities 2135
AIM EXPRESS-G entity (figure H.115) 2534
Point_on_surface
application assertion 524, 539
application object 278
ARM EXPRESS-G (figure G.43) 2410
mapping table 888, 932, 968
point_on_surface
AIM expanded listing entities 2135
AIM EXPRESS-G entity (figure H.58) 2477
mapping table 873, 888, 916, 932, 962, 968
point_on_surface_pair
AIM expanded listing entities 2135
AIM EXPRESS-G entity (figure H.117) 2536
point_on_surface_pair_range
AIM expanded listing entities 2135
AIM EXPRESS-G entity (figure H.121) 2540
point_on_surface_pair_value
AIM expanded listing entities 2136
AIM EXPRESS-G entity (figure H.113) 2532
point_placement_shape_representation
AIM expanded listing entities 2136
AIM EXPRESS-G entity (figure H.20) 2439
AIM short listing entities 1654
mapping table 982
point_replica
AIM expanded listing entities 2136
AIM EXPRESS-G entity (figure H.58) 2477
mapping table 866
Point_sequence
application assertion 539, 540
application object 278
ARM EXPRESS-G (figure G.42) 2409
mapping table 983
point_style
AIM expanded listing entities 2136
AIM EXPRESS-G entity (figure H.94) 2513
mapping table 990, 991
Poly_loop
application assertion 488, 540

application object 278
ARM EXPRESS-G (figure G.45) 2412
mapping table 897, 958
poly_loop
AIM expanded listing entities 2137
AIM EXPRESS-G entity (figure H.81) 2500
Polyline
application assertion 528
application object 279
ARM EXPRESS-G (figure G.43) 2410
mapping table 865, 872, 888, 914, 932, 958, 968
polyline
AIM expanded listing entities 2137
AIM EXPRESS-G entity (figure H.63) 2482
mapping table 932
Position_tolerance
application assertion 540
application object 279
ARM EXPRESS-G (figure G.48) 2415
mapping table 1457
position_tolerance
AIM expanded listing entities 2137
AIM EXPRESS-G entity (figure H.110) 2529
mapping table 1457
positive_length_measure
AIM expanded listing types 1994
AIM EXPRESS-G types (figure H.48) 2467
positive_plane_angle_measure
AIM expanded listing types 1995
AIM EXPRESS-G types (figure H.48) 2467
positive_ratio_measure
AIM expanded listing types 1995
AIM EXPRESS-G types (figure H.48) 2467
pre_defined_colour
AIM expanded listing entities 2137
AIM EXPRESS-G entity (figure H.91) 2510
AIM short listing imported entity modifications 1755
pre_defined_curve_font
AIM expanded listing entities 2137
AIM EXPRESS-G entity (figure H.55) 2474
AIM short listing imported entity modifications 1755
pre_defined_item
AIM expanded listing entities 2137
AIM EXPRESS-G entity (figure H.55) 2474
pre_defined_marker
AIM expanded listing entities 2137
AIM EXPRESS-G entity (figure H.55) 2474
pre_defined_point_marker_symbol
AIM expanded listing entities 2137
AIM EXPRESS-G entity (figure H.55) 2474
AIM short listing entities 1655
pre_defined_symbol

AIM expanded listing entities 2137
AIM EXPRESS-G entity (figure H.55) 2474
AIM short listing imported entity modifications 1755
pre_defined_terminator_symbol
AIM expanded listing entities 2137
AIM EXPRESS-G entity (figure H.55) 2474
AIM short listing entities 1656
mapping table 997
pre_defined_text_font
AIM expanded listing entities 2138
AIM EXPRESS-G entity (figure H.55) 2474
AIM short listing imported entity modifications 1756
precision_qualifier
AIM expanded listing entities 2138
AIM EXPRESS-G entity (figure H.56) 2475
mapping table 1430, 1449
preferred_surface_curve_representation
AIM expanded listing types 1995
AIM EXPRESS-G types (figure H.66) 2485
presentable_text
AIM expanded listing types 1995
AIM EXPRESS-G types (figure H.89) 2508
presentation_area
AIM expanded listing entities 2138
AIM EXPRESS-G entity (figure H.90) 2509
AIM short listing imported entity modifications 1756
mapping table 1001
presentation_layer_assignment
AIM expanded listing entities 2138
AIM EXPRESS-G entity (figure H.90) 2509
AIM short listing imported entity modifications 1756
mapping table 1168, 1169, 1179, 1180
presentation_layer_assignment_constraint_2d_or_3d
AIM expanded listing rules 2265
AIM short listing rules 1810
mapping table 1169
presentation_layer_usage
AIM expanded listing entities 2138
presentation_representation
AIM expanded listing entities 2138
AIM EXPRESS-G entity (figure H.90) 2509
presentation_representation_relationship
AIM EXPRESS-G entity (figure H.19) 2438
presentation_representation_select
AIM expanded listing types 1995
AIM EXPRESS-G types (figure H.90) 2509
presentation_set
AIM expanded listing entities 2138
AIM EXPRESS-G entity (figure H.90) 2509
presentation_size
AIM expanded listing entities 2139
AIM EXPRESS-G entity (figure H.90) 2509

mapping table 631, 1002
presentation_size_assignment_select
AIM expanded listing types 1995
AIM EXPRESS-G types (figure H.90) 2509
presentation_style_assignment
AIM expanded listing entities 2139
AIM EXPRESS-G entity (figure H.93) 2512
mapping table 984
presentation_style_by_context
AIM expanded listing entities 2139
AIM EXPRESS-G entity (figure H.93) 2512
AIM short listing imported entity modifications 1756
mapping table 985
presentation_style_select
AIM expanded listing types 1995
AIM EXPRESS-G types (figure H.93) 2512
presentation_view
AIM expanded listing entities 2139
AIM EXPRESS-G entity (figure H.91) 2510
AIM short listing imported entity modifications 1757
mapping table 997
presentation_view_relationship
AIM expanded listing entities 2139
AIM EXPRESS-G entity (figure H.19) 2438
AIM short listing entities 1658
mapping table 1013
presented_item
AIM expanded listing entities 2139
AIM EXPRESS-G entity (figure H.90) 2509
presented_item_representation
AIM expanded listing entities 2140
AIM EXPRESS-G entity (figure H.90) 2509
presented_item_select
AIM expanded listing types 1995
AIM EXPRESS-G types (figure H.99) 2518
AIM short listing types 1516
prismatic_pair
AIM expanded listing entities 2140
AIM EXPRESS-G entity (figure H.116) 2535
prismatic_pair_range
AIM expanded listing entities 2140
AIM EXPRESS-G entity (figure H.121) 2540
prismatic_pair_value
AIM expanded listing entities 2140
AIM EXPRESS-G entity (figure H.113) 2532
process_change
AIM expanded listing entities 2140
AIM EXPRESS-G entity (figure H.102) 2521
AIM short listing entities 1658
mapping table 1169
Process_cost_property
application assertion 540

application object 281
ARM EXPRESS-G (figure G.20) 2387
mapping table 1394
Process_feature_in_panel
application assertion 433
application object 281
ARM EXPRESS-G (figure G.28) 2395
mapping table 791
Process_feature_in_solid
application assertion 434
application object 281
ARM EXPRESS-G (figure G.28) 2395
mapping table 792
Process_operation
application assertion 436, 466, 481, 484, 540–545, 573, 580, 589
application object 282
ARM EXPRESS-G (figure G.19) 2386
mapping table 1360, 1394
Process_operation_input_or_output
application assertion 541, 542
application object 283
ARM EXPRESS-G (figure G.19) 2386
mapping table 1397
Process_operation_relationship
application assertion 427, 542, 543
application object 285
ARM EXPRESS-G (figure G.19) 2386
mapping table 1405
Process_operation_resource
application assertion 484, 543
application object 287
ARM EXPRESS-G (figure G.19) 2386
mapping table 1406
Process_operation_resource_assignment
application assertion 543, 544
application object 287
ARM EXPRESS-G (figure G.19) 2386
mapping table 1407
process_or_process_relationship
AIM expanded listing types 1995
AIM EXPRESS-G types (figure H.124) 2543
Process_plan
application assertion 402, 414, 446, 453, 466, 472, 479, 485, 534, 544, 545, 558, 566, 589, 596
application object 289
ARM EXPRESS-G (figure G.19) 2386
mapping table 1410
process_plan
AIM expanded listing entities 2141
AIM EXPRESS-G entity (figure H.102) 2521
AIM short listing entities 1660
mapping table 1410, 1413
process_plan_assignment

AIM expanded listing entities 2141
AIM EXPRESS-G entity (figure H.41) 2460
AIM short listing entities 1661
process_plan_item
AIM expanded listing types 1996
AIM EXPRESS-G types (figure H.41) 2460
AIM short listing types 1517
Process_plan_operation_assignment
application assertion 542, 544, 545
application object 290
ARM EXPRESS-G (figure G.19) 2386
mapping table 1412
Process_plan_relationship
application assertion 427, 545
application object 291
ARM EXPRESS-G (figure G.19) 2386
mapping table 1412
Process_plan_version
application object 292
ARM EXPRESS-G (figure G.19) 2386
mapping table 1413
process_product_association
AIM expanded listing entities 2142
AIM EXPRESS-G entity (figure H.102) 2521
AIM short listing imported entity modifications 1757
mapping table 1397, 1399
Process_property
application assertion 545
application object 292
ARM EXPRESS-G (figure G.20) 2387
mapping table 1414
process_property_association
AIM expanded listing entities 2142
AIM EXPRESS-G entity (figure H.103) 2522
Process_quality_property
application assertion 545, 546
application object 292
ARM EXPRESS-G (figure G.20) 2387
mapping table 1414
Process_state
application assertion 546
application object 293
ARM EXPRESS-G (figure G.5) 2372
mapping table 1415
product
AIM expanded listing entities 2142
AIM EXPRESS-G entity (figure H.4) 2423
AIM short listing imported entity modifications 1757
mapping table 1121, 1122
product_category
AIM expanded listing entities 2142
AIM EXPRESS-G entity (figure H.2) 2421

AIM short listing imported entity modifications 1757
mapping table 1122, 1158, 1159, 1333, 1335, 1336
product_category_relationship
AIM expanded listing entities 2142
AIM EXPRESS-G entity (figure H.2) 2421
mapping table 1335
Product_class
application assertion 402, 407, 414, 429, 430, 446, 453, 460, 466, 472, 479, 485, 510, 522, 535, 546, 548, 550, 559, 566, 572, 589, 596
application object 293
ARM EXPRESS-G (figure G.6) 2373
mapping table 1362
product_class
AIM expanded listing entities 2142
AIM EXPRESS-G entity (figure H.2) 2421
AIM short listing entities 1662
mapping table 1362
Product_class_hierarchy
application assertion 546
application object 295
ARM EXPRESS-G (figure G.6) 2373
mapping table 1363
product_class_hierarchy
AIM expanded listing functions 2241
AIM short listing functions 1937
Product_component
application assertion 403, 407, 414, 446, 453, 461, 466, 472, 479, 485, 491, 505, 510, 512, 522, 535, 546, 547, 559, 566, 572, 589, 596
application object 296
ARM EXPRESS-G (figure G.7) 2374
mapping table 1364
product_component
AIM expanded listing entities 2143
AIM EXPRESS-G entity (figure H.2) 2421
AIM short listing entities 1664
mapping table 1364
Product_component_relationship
application assertion 427, 547
application object 298
ARM EXPRESS-G (figure G.7) 2374
mapping table 1366
product_concept
AIM expanded listing entities 2143
AIM EXPRESS-G entity (figure H.5) 2424
AIM short listing imported entity modifications 1758
mapping table 1362
product_concept_context
AIM expanded listing entities 2143
AIM EXPRESS-G entity (figure H.1) 2420
product_concept_feature
AIM expanded listing entities 2143
AIM EXPRESS-G entity (figure H.7) 2426

AIM short listing imported entity modifications 1758
mapping table 1376, 1378, 1380
product_concept_feature_association
AIM expanded listing entities 2143
AIM EXPRESS-G entity (figure H.7) 2426
AIM short listing imported entity modifications 1758
mapping table 1337, 1339, 1340
product_concept_feature_category
AIM expanded listing entities 2143
AIM EXPRESS-G entity (figure H.52) 2471
AIM short listing entities 1664
mapping table 1378
product_concept_feature_category_usage
AIM expanded listing entities 2144
AIM EXPRESS-G entity (figure H.52) 2471
AIM short listing entities 1665
mapping table 1337
product_concept_relationship
AIM expanded listing entities 2144
AIM EXPRESS-G entity (figure H.5) 2424
mapping table 1363
product_context
AIM expanded listing entities 2144
AIM EXPRESS-G entity (figure H.1) 2420
product_data_representation_view
AIM EXPRESS-G entity (figure H.90) 2509
product_definition
AIM expanded listing entities 2144
AIM EXPRESS-G entity (figure H.4) 2423
AIM short listing imported entity modifications 1758
mapping table 1120, 1121, 1190, 1200, 1342, 1354, 1357, 1367, 1368, 1383, 1393
product_definition_context
AIM expanded listing entities 2144
AIM EXPRESS-G entity (figure H.1) 2420
mapping table 1060, 1061, 1209
product_definition_formation
AIM expanded listing entities 2144
AIM EXPRESS-G entity (figure H.4) 2423
AIM short listing imported entity modifications 1759
mapping table 1123
product_definition_formation_relationship
AIM expanded listing entities 2145
AIM EXPRESS-G entity (figure H.5) 2424
AIM short listing imported entity modifications 1759
mapping table 1290, 1292, 1298
product_definition_occurrence
AIM expanded listing entities 2145
AIM EXPRESS-G entity (figure H.4) 2423
AIM short listing entities 1666
mapping table 1199, 1203, 1204, 1206
product_definition_process
AIM expanded listing entities 2146

AIM EXPRESS-G entity (figure H.102) 2521
product_definition_relationship
AIM expanded listing entities 2146
AIM EXPRESS-G entity (figure H.5) 2424
AIM short listing imported entity modifications 1759
mapping table 652, 653, 1191–1194, 1198–1200, 1202, 1290, 1292, 1341, 1355, 1356, 1369, 1370, 1372, 1376, 1385, 1386, 1415, 1420
product_definition_resource
AIM expanded listing entities 2146
AIM EXPRESS-G entity (figure H.4) 2423
AIM short listing entities 1669
product_definition_shape
AIM expanded listing entities 2146
AIM EXPRESS-G entity (figure H.10) 2429
AIM short listing imported entity modifications 1760
mapping table 854, 1167
product_definition_substitute
AIM expanded listing entities 2147
AIM EXPRESS-G entity (figure H.5) 2424
AIM short listing imported entity modifications 1760
mapping table 1202, 1373
product_definition_substitute_requires_effectivity_assignment
AIM expanded listing rules 2266
AIM short listing rules 1811
mapping table 1219
product_definition_subtype_exclusiveness
AIM expanded listing rules 2266
AIM short listing rules 1812
product_definition_usage
AIM expanded listing entities 2147
AIM EXPRESS-G entity (figure H.6) 2425
product_definition_with_associated_documents
AIM expanded listing entities 2147
AIM EXPRESS-G entity (figure H.4) 2423
AIM short listing imported entity modifications 1760
Product_function
application assertion 403, 407, 415, 430, 447, 453, 467, 472, 479, 485, 491, 510, 522, 535, 547, 548, 559, 566, 573, 589, 596
application object 299
ARM EXPRESS-G (figure G.7) 2374
mapping table 1367
Product_function_relationship
application assertion 427, 548
application object 300
ARM EXPRESS-G (figure G.7) 2374
mapping table 1369
product_related_product_category
AIM expanded listing entities 2147
AIM EXPRESS-G entity (figure H.2) 2421
AIM short listing imported entity modifications 1760
product_requires_version
AIM expanded listing rules 2266

AIM short listing rules 1813
mapping table 1121
Product_specification
application assertion 403, 447, 454, 472, 479, 485, 508, 535, 537, 548, 559, 596
application object 301
ARM EXPRESS-G (figure G.9) 2376
mapping table 1370
Profile
application assertion 496, 548, 549
application object 302
ARM EXPRESS-G (figure G.30) 2397
mapping table 792
Project
application assertion 415, 447, 454, 479, 485, 535, 549, 550, 559, 567
application object 304
ARM EXPRESS-G (figure G.17) 2384
mapping table 1301
project
AIM expanded listing entities 2147
AIM EXPRESS-G entity (figure H.24) 2443
AIM short listing entities 1669
mapping table 1301
project_assignment
AIM expanded listing entities 2148
AIM EXPRESS-G entity (figure H.24) 2443
AIM short listing entities 1672
project_item
AIM expanded listing types 1996
AIM EXPRESS-G types (figure H.24) 2443
AIM short listing types 1517
Project_relationship
application assertion 550
application object 305
ARM EXPRESS-G (figure G.17) 2384
mapping table 1308
projected_zone_definition
AIM expanded listing entities 2148
AIM EXPRESS-G entity (figure H.109) 2528
mapping table 1458
Projection
application assertion 550
application object 307
ARM EXPRESS-G (figure G.49) 2416
mapping table 1458
Projection_curve
application object 307
ARM EXPRESS-G (figure G.34) 2401
mapping table 633
projection_curve
AIM expanded listing entities 2148
AIM EXPRESS-G entity (figure H.85) 2504
mapping table 644

projection_directed_callout
AIM expanded listing entities 2149
AIM EXPRESS-G entity (figure H.106) 2525
Promissory_usage
application object 307
ARM EXPRESS-G (figure G.13) 2380
mapping table 1203
promissory_usage_occurrence
AIM expanded listing entities 2149
AIM EXPRESS-G entity (figure H.6) 2425
mapping table 1203
Property
application assertion 403, 447, 454, 472, 479, 486, 535, 551, 559, 567, 597
application object 308
ARM EXPRESS-G (figure G.23) 2390
mapping table 1039
Property_change
application assertion 550, 551
application object 308
ARM EXPRESS-G (figure G.18) 2385
mapping table 1039
property_change
AIM expanded listing entities 2149
AIM EXPRESS-G entity (figure H.10) 2429
AIM short listing entities 1672
mapping table 1039
property_definition
AIM expanded listing entities 2149
AIM EXPRESS-G entity (figure H.10) 2429
AIM short listing imported entity modifications 1761
mapping table 1021, 1024, 1026, 1030, 1035, 1037, 1039, 1042, 1043, 1167, 1169
property_definition_relationship
AIM expanded listing entities 2149
AIM EXPRESS-G entity (figure H.10) 2429
AIM short listing imported entity modifications 1761
mapping table 1040
property_definition_representation
AIM expanded listing entities 2149
AIM EXPRESS-G entity (figure H.13) 2432
AIM short listing imported entity modifications 1761
mapping table 607, 733, 1022, 1176, 1347, 1428
property_or_shape_select
AIM expanded listing types 1996
AIM EXPRESS-G types (figure H.103) 2522
property_process
AIM expanded listing entities 2150
AIM EXPRESS-G entity (figure H.103) 2522
Property_relationship
application assertion 551
application object 309
ARM EXPRESS-G (figure G.23) 2390
mapping table 1040

Property_value
application assertion 431, 496, 497, 506, 511, 519, 524, 540, 541, 543, 544, 546, 551, 553, 583
application object ... 310
ARM EXPRESS-G (figure G.23) ... 2390
mapping table ... 1041
property_value_representation
AIM expanded listing entities ... 2150
AIM EXPRESS-G entity (figure H.16) ... 2435
AIM short listing entities ... 1673
mapping table ... 1041

qualified_representation_item
AIM expanded listing entities ... 2150
AIM EXPRESS-G entity (figure H.56) ... 2475
mapping table ... 1161, 1162, 1439
qualitative_uncertainty
AIM expanded listing entities ... 2150
mapping table ... 1161, 1162
quantified_assembly_component_usage
AIM expanded listing entities ... 2150
AIM EXPRESS-G entity (figure H.6) ... 2425
mapping table ... 1203
Quantified_instance
application assertion ... 520, 552
application object ... 311
ARM EXPRESS-G (figure G.14) ... 2381
mapping table ... 1203
quasi_uniform_curve
AIM expanded listing entities ... 2150
AIM EXPRESS-G entity (figure H.64) ... 2483
quasi_uniform_surface
AIM expanded listing entities ... 2150
AIM EXPRESS-G entity (figure H.72) ... 2491

rack_and_pinion_pair
AIM expanded listing entities ... 2151
AIM EXPRESS-G entity (figure H.118) ... 2537
rack_and_pinion_pair_range
AIM expanded listing entities ... 2151
AIM EXPRESS-G entity (figure H.121) ... 2540
rack_and_pinion_pair_value
AIM expanded listing entities ... 2151
AIM EXPRESS-G entity (figure H.113) ... 2532
radius_dimension
AIM expanded listing entities ... 2151
AIM EXPRESS-G entity (figure H.106) ... 2525
mapping table ... 622
Radiused_bead_end_type
application object ... 311
ARM EXPRESS-G (figure G.38) ... 2405
mapping table ... 794
Radiused_slot_end_type
application object ... 311

ARM EXPRESS-G (figure G.36) ... 2403
mapping table ... 794
ratio_measure
AIM expanded listing types ... 1996
AIM EXPRESS-G types (figure H.48) ... 2467
ratio_measure_with_unit
AIM expanded listing entities ... 2151
AIM EXPRESS-G entity (figure H.46) ... 2465
ratio_unit
AIM expanded listing entities ... 2151
rational_b_spline_curve
AIM expanded listing entities ... 2152
AIM EXPRESS-G entity (figure H.64) ... 2483
rational_b_spline_surface
AIM expanded listing entities ... 2152
AIM EXPRESS-G entity (figure H.72) ... 2491
Rectangular_boss
application assertion ... 552
application object ... 312
ARM EXPRESS-G (figure G.37) ... 2404
mapping table ... 795
Rectangular_closed_pocket_in_solid
application assertion ... 552
application object ... 312
ARM EXPRESS-G (figure G.37) ... 2404
mapping table ... 795
Rectangular_closed_profile
application assertion ... 552
application object ... 313
ARM EXPRESS-G (figure G.30) ... 2397
mapping table ... 796
rectangular_closed_profile
AIM expanded listing entities ... 2152
AIM EXPRESS-G entity (figure H.11) ... 2430
AIM short listing entities ... 1674
mapping table ... 708, 792, 796
rectangular_composite_surface
AIM expanded listing entities ... 2153
AIM EXPRESS-G entity (figure H.73) ... 2492
mapping table ... 969
Rectangular_offset_pattern
application assertion ... 552, 553
application object ... 313
ARM EXPRESS-G (figure G.40) ... 2407
mapping table ... 798
Rectangular_omit_pattern
application assertion ... 553
application object ... 315
ARM EXPRESS-G (figure G.40) ... 2407
mapping table ... 801
Rectangular_open_pocket_in_solid
application assertion ... 552

application object 315
ARM EXPRESS-G (figure G.37) 2404
mapping table 803
Rectangular_pattern
application assertion 553
application object 316
ARM EXPRESS-G (figure G.40) 2407
mapping table 803
rectangular_pattern
AIM expanded listing entities 2153
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1677
mapping table 803
rectangular_trimmed_surface
AIM expanded listing entities 2155
AIM EXPRESS-G entity (figure H.73) 2492
mapping table 969, 971
Recyclability
application assertion 553
application object 317
ARM EXPRESS-G (figure G.23) 2390
mapping table 1042
Reference_plane
application assertion 517, 529, 540, 578, 581
application object 319
ARM EXPRESS-G (figure G.48) 2415
mapping table 1459
referenced_modified_datum
AIM expanded listing entities 2155
AIM EXPRESS-G entity (figure H.110) 2529
relationship_condition
AIM expanded listing entities 2155
AIM EXPRESS-G entity (figure H.38) 2457
mapping table 1405
relationship_with_condition
AIM expanded listing types 1996
AIM EXPRESS-G types (figure H.38) 2457
rendering_properties_select
AIM expanded listing types 1996
AIM EXPRESS-G types (figure H.95) 2514
rep_item_group
AIM expanded listing entities 2155
AIM EXPRESS-G entity (figure H.52) 2471
AIM short listing entities 1679
mapping table 1165
reparametrised_composite_curve_segment
AIM expanded listing entities 2156
AIM EXPRESS-G entity (figure H.65) 2484
Replaced_definition_relationship
application assertion 427, 553
application object 319
ARM EXPRESS-G (figure G.13) 2380

mapping table 1372
Replaced_usage_relationship
application assertion 427, 553
application object 320
ARM EXPRESS-G (figure G.14) 2381
mapping table 1373
Replicate_base_assignment
application assertion 554
application object 320
ARM EXPRESS-G (figure G.40) 2407
mapping table 809
Replicate_feature
application assertion 554
application object 321
ARM EXPRESS-G (figure G.40) 2407
mapping table 811
replicate_feature
AIM expanded listing entities 2156
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1680
mapping table 811
representation
AIM expanded listing entities 2156
AIM EXPRESS-G entity (figure H.16) 2435
AIM short listing imported entity modifications 1761
mapping table 602, 605, 606, 609, 611–613, 615–619, 632, 668, 729–732, 862, 869, 883, 894, 896, 909, 911, 927, 929, 953, 955, 965, 992, 997, 998, 1001, 1024, 1036, 1041, 1329, 1346, 1349, 1430, 1433, 1441
representation_context
AIM expanded listing entities 2156
AIM EXPRESS-G entity (figure H.19) 2438
mapping table 613, 1021, 1037, 1042, 1431, 1450
representation_item
AIM expanded listing entities 2156
AIM EXPRESS-G entity (figure H.17) 2436
AIM short listing imported entity modifications 1762
mapping table . 621, 622, 647, 676, 859, 861, 862, 866, 868, 869, 873, 877, 890, 892, 893, 899, 903, 916, 920, 921, 939, 945, 962, 988, 1007, 1180, 1427
representation_item_relationship
AIM expanded listing entities 2156
representation_map
AIM expanded listing entities 2157
AIM EXPRESS-G entity (figure H.17) 2436
mapping table 1181, 1182, 1184, 1190
representation_of_link
AIM expanded listing functions 2242
representation_relationship
AIM expanded listing entities 2157
AIM EXPRESS-G entity (figure H.18) 2437
AIM short listing imported entity modifications 1762
mapping table . 607, 609, 610, 733, 860, 863, 867, 870, 875, 884, 891, 895, 901, 910, 919, 928, 942, 954, 964, 966, 993, 1010, 1011, 1013, 1022, 1290, 1292, 1348, 1428

representation_relationship_with_transformation
AIM expanded listing entities 2157
AIM EXPRESS-G entity (figure H.18) 2437
mapping table 1164, 1352
representation_subtype_exclusiveness
AIM expanded listing rules 2266
AIM short listing rules 1814
mapping table 668, 991
representations_mapped_into
AIM expanded listing functions 2242
represented_definition
AIM expanded listing types 1996
AIM EXPRESS-G types (figure H.13) 2432
requirement_for_action_resource
AIM expanded listing entities 2157
AIM EXPRESS-G entity (figure H.40) 2459
resource_property
AIM expanded listing entities 2157
AIM EXPRESS-G entity (figure H.40) 2459
resource_property_representation
AIM expanded listing entities 2157
AIM EXPRESS-G entity (figure H.40) 2459
resource_requirement_type
AIM expanded listing entities 2157
AIM EXPRESS-G entity (figure H.40) 2459
AIM short listing imported entity modifications 1762
mapping table 1406
restrict_action_directive
AIM expanded listing rules 2267
AIM short listing rules 1816
restrict_action_resource_requirement_for_process_operation
AIM expanded listing rules 2267
AIM short listing rules 1817
mapping table 1407
restrict_approval_date_time_for_approval
AIM expanded listing rules 2268
AIM short listing rules 1818
mapping table 1061, 1075
restrict_approval_date_time_role
AIM expanded listing rules 2268
AIM short listing rules 1819
restrict_classification_system_for_approval_status
AIM expanded listing rules 2268
AIM short listing rules 1819
mapping table 1076
restrict_classification_system_for_document_type
AIM expanded listing rules 2268
AIM short listing rules 1820
restrict_classification_system_for_security_classification_level
AIM expanded listing rules 2268
AIM short listing rules 1821
restrict_multi_language_for_action

AIM expanded listing rules 2269
AIM short listing rules 1822
mapping table 1269, 1361, 1396
restrict_multi_language_for_action_directive
AIM expanded listing rules 2269
AIM short listing rules 1823
mapping table 1310
restrict_multi_language_for_action_method
AIM expanded listing rules 2269
AIM short listing rules 1824
mapping table 1288, 1360, 1395
restrict_multi_language_for_action_property
AIM expanded listing rules 2270
AIM short listing rules 1825
mapping table 1384, 1414
restrict_multi_language_for_action_relationship
AIM expanded listing rules 2270
AIM short listing rules 1826
mapping table 1289, 1295, 1309, 1405, 1413
restrict_multi_language_for_action_resource_requirement
AIM expanded listing rules 2270
AIM short listing rules 1827
mapping table 1408
restrict_multi_language_for_alternate_product_relationship
AIM expanded listing rules 2270
AIM short listing rules 1828
mapping table 1290
restrict_multi_language_for_application_context
AIM expanded listing rules 2271
AIM short listing rules 1829
mapping table 1061
restrict_multi_language_for_approval
AIM expanded listing rules 2271
AIM short listing rules 1830
mapping table 1062
restrict_multi_language_for_assembly_component_usage
AIM expanded listing rules 2271
AIM short listing rules 1831
mapping table 1201
restrict_multi_language_for_assembly_component_usage_substitute
AIM expanded listing rules 2272
AIM short listing rules 1832
mapping table 1191
restrict_multi_language_for_camera_image_2d_with_scale
AIM expanded listing rules 2272
AIM short listing rules 1833
mapping table 990
restrict_multi_language_for_camera_image_3d_with_scale
AIM expanded listing rules 2272
AIM short listing rules 1834
mapping table 990
restrict_multi_language_for_certification

AIM expanded listing rules 2273
AIM short listing rules 1835
mapping table 1375
restrict_multi_language_for_characterized_product_category
AIM expanded listing rules 2273
AIM short listing rules 1836
mapping table 1334
restrict_multi_language_for_classification_system
AIM expanded listing rules 2273
AIM short listing rules 1837
mapping table 1332
restrict_multi_language_for_configurable_item
AIM expanded listing rules 2274
AIM short listing rules 1838
mapping table 1371
restrict_multi_language_for_configuration_design
AIM expanded listing rules 2274
AIM short listing rules 1839
mapping table 1338
restrict_multi_language_for_configuration_item_relationship
AIM expanded listing rules 2274
AIM short listing rules 1840
mapping table 1295, 1367
restrict_multi_language_for_constructive_geometry_representation_relationship
AIM expanded listing rules 2275
AIM short listing rules 1841
mapping table 861, 868, 876, 892, 902, 920, 942, 964
restrict_multi_language_for_data_environment
AIM expanded listing rules 2275
AIM short listing rules 1842
mapping table 1022
restrict_multi_language_for_default_tolerance_table_cell
AIM expanded listing rules 2275
AIM short listing rules 1843
mapping table 1442
restrict_multi_language_for_descriptive_representation_item
AIM expanded listing rules 2276
AIM short listing rules 1844
mapping table 607, 610, 616–618, 620, 729–732, 1042, 1330, 1345, 1346
restrict_multi_language_for_dimensional_location
AIM expanded listing rules 2276
AIM short listing rules 1845
mapping table 1435
restrict_multi_language_for_dimensional_size
AIM expanded listing rules 2276
AIM short listing rules 1846
mapping table 1435
restrict_multi_language_for_document
AIM expanded listing rules 2277
AIM short listing rules 1847
mapping table 657
restrict_multi_language_for_document_relationship

AIM expanded listing rules 2277
AIM short listing rules 1848
mapping table 653
restrict_multi_language_for_draughting_model
AIM expanded listing rules 2277
AIM short listing rules 1849
mapping table 992
restrict_multi_language_for_draughting_title
AIM expanded listing rules 2277
AIM short listing rules 1850
mapping table 627
restrict_multi_language_for_drawing_sheet_revision
AIM expanded listing rules 2278
AIM short listing rules 1850
mapping table 632
restrict_multi_language_for_effectivity
AIM expanded listing rules 2278
AIM short listing rules 1851
mapping table 1210
restrict_multi_language_for_effectivity_relationship
AIM expanded listing rules 2278
AIM short listing rules 1852
mapping table 1226
restrict_multi_language_for_externally_defined_classification_attribute
AIM short listing rules 1853
mapping table 1331
restrict_multi_language_for_externally_defined_classification_attribute
AIM expanded listing rules 2279
restrict_multi_language_for_externally_defined_classification_system
AIM expanded listing rules 2279
AIM short listing rules 1854
mapping table 1332
restrict_multi_language_for_externally_defined_product_category
AIM expanded listing rules 2279
AIM short listing rules 1855
mapping table 1334
restrict_multi_language_for_externally_defined_representation
AIM expanded listing rules 2280
AIM short listing rules 1856
mapping table 668
restrict_multi_language_for_geometric_tolerance
AIM expanded listing rules 2280
AIM short listing rules 1857
mapping table 1452
restrict_multi_language_for_group
AIM expanded listing rules 2280
AIM short listing rules 1858
mapping table 1165, 1166
restrict_multi_language_for_identification_role
AIM expanded listing rules 2281
AIM short listing rules 1859
restrict_multi_language_for_make_from_usage_option

AIM expanded listing rules 2281
AIM short listing rules 1860
mapping table 1385, 1386
restrict_multi_language_for_material_property
AIM expanded listing rules 2281
AIM short listing rules 1861
mapping table 1035
restrict_multi_language_for_measure_representation_item
AIM expanded listing rules 2282
AIM short listing rules 1862
mapping table 676
restrict_multi_language_for_mechanical_design_geometric_presentation_representation
AIM expanded listing rules 2282
AIM short listing rules 1863
mapping table 992
restrict_multi_language_for_model_change
AIM expanded listing rules 2282
AIM short listing rules 1864
mapping table 1174
restrict_multi_language_for_organization_relationship
AIM expanded listing rules 2283
AIM short listing rules 1865
mapping table 1308
restrict_multi_language_for_organization_role
AIM expanded listing rules 2283
AIM short listing rules 1866
mapping table 1299
restrict_multi_language_for_pair_actuator
AIM expanded listing rules 2283
AIM short listing rules 1867
mapping table 970
restrict_multi_language_for_presentation_area
AIM expanded listing rules 2284
AIM short listing rules 1868
mapping table 1002, 1003
restrict_multi_language_for_presentation_layer_assignment
AIM expanded listing rules 2284
AIM short listing rules 1869
mapping table 1169
restrict_multi_language_for_presentation_view
AIM expanded listing rules 2284
AIM short listing rules 1870
mapping table 998, 999
restrict_multi_language_for_presentation_view_relationship
AIM expanded listing rules 2285
AIM short listing rules 1871
mapping table 1014
restrict_multi_language_for_process_change
AIM expanded listing rules 2285
AIM short listing rules 1872
mapping table 1174
restrict_multi_language_for_process_plan

AIM expanded listing rules 2285
AIM short listing rules 1873
mapping table 1411
restrict_multi_language_for_process_product_association
AIM expanded listing rules 2286
AIM short listing rules 1874
mapping table 1399
restrict_multi_language_for_product
AIM expanded listing rules 2286
AIM short listing rules 1875
mapping table 1122
restrict_multi_language_for_product_category
AIM expanded listing rules 2286
AIM short listing rules 1876
mapping table 1123
restrict_multi_language_for_product_class
AIM expanded listing rules 2286
AIM short listing rules 1877
mapping table 1363
restrict_multi_language_for_product_component
AIM expanded listing rules 2287
AIM short listing rules 1878
mapping table 1365, 1366
restrict_multi_language_for_product_concept_feature
AIM expanded listing rules 2287
AIM short listing rules 1879
mapping table 1377, 1379, 1380
restrict_multi_language_for_product_concept_feature_association
AIM expanded listing rules 2287
AIM short listing rules 1880
mapping table 1338, 1340
restrict_multi_language_for_product_concept_feature_category
AIM expanded listing rules 2288
AIM short listing rules 1881
mapping table 1378
restrict_multi_language_for_product_definition
AIM expanded listing rules 2288
AIM short listing rules 1882
mapping table 1121, 1342
restrict_multi_language_for_product_definition_formation
AIM expanded listing rules 2288
AIM short listing rules 1883
mapping table 1123
restrict_multi_language_for_product_definition_formation_relationship
AIM expanded listing rules 2289
AIM short listing rules 1884
mapping table 1295, 1298
restrict_multi_language_for_product_definition_occurrence
AIM expanded listing rules 2289
AIM short listing rules 1885
mapping table 1201
restrict_multi_language_for_product_definition_relationship

AIM expanded listing rules 2289
AIM short listing rules 1886
mapping table 1192–1194, 1201, 1356, 1370, 1373, 1415
restrict_multi_language_for_product_definition_shape
AIM expanded listing rules 2290
AIM short listing rules 1887
mapping table 1168
restrict_multi_language_for_product_definition_substitute
AIM expanded listing rules 2290
AIM short listing rules 1888
mapping table 1373
restrict_multi_language_for_project
AIM expanded listing rules 2290
AIM short listing rules 1889
mapping table 1303
restrict_multi_language_for_property_change
AIM expanded listing rules 2291
AIM short listing rules 1890
mapping table 1039, 1040
restrict_multi_language_for_property_definition
AIM expanded listing rules 2291
AIM short listing rules 1891
mapping table 1021, 1025, 1030, 1043
restrict_multi_language_for_property_definition_relationship
AIM expanded listing rules 2291
AIM short listing rules 1892
restrict_multi_language_for_property_definition_representation
AIM expanded listing rules 2292
AIM short listing rules 1893
mapping table 1348
restrict_multi_language_for_rep_item_group
AIM expanded listing rules 2292
AIM short listing rules 1894
mapping table 1165, 1166
restrict_multi_language_for_representation
AIM expanded listing rules 2292
AIM short listing rules 1895
mapping table 616, 618, 620, 1024, 1331, 1347
restrict_multi_language_for_representation_item
AIM expanded listing rules 2293
AIM short listing rules 1896
mapping table 621, 647, 862, 869, 877, 892, 903, 920, 945, 1007
restrict_multi_language_for_representation_relationship
AIM expanded listing rules 2293
AIM short listing rules 1897
mapping table 863, 870, 884, 895, 910, 928, 954, 966, 993, 1295, 1331, 1349
restrict_multi_language_for_resource_requirement_type
AIM expanded listing rules 2293
AIM short listing rules 1898
mapping table 1406
restrict_multi_language_for_retention
AIM expanded listing rules 2293

AIM short listing rules 1899
mapping table 1265
restrict_multi_language_for_security_classification
AIM expanded listing rules 2294
AIM short listing rules 1900
mapping table 1157
restrict_multi_language_for_shape_aspect
AIM expanded listing rules 2294
AIM short listing rules 1901
mapping table 672, 745, 849, 1175
restrict_multi_language_for_shape_aspect_relationship
AIM expanded listing rules 2294
AIM short listing rules 1902
mapping table 604, 605, 1176, 1447
restrict_multi_language_for_shape_dimension_representation
AIM expanded listing rules 2295
AIM short listing rules 1903
mapping table 1433
restrict_multi_language_for_shape_representation
AIM expanded listing rules 2295
AIM short listing rules 1904
mapping table 863, 870, 884, 895, 910, 927, 953, 965
restrict_multi_language_for_solution_based_product_definition
AIM expanded listing rules 2295
AIM short listing rules 1905
mapping table 1383
restrict_multi_language_for_versioned_action_request
AIM expanded listing rules 2296
AIM short listing rules 1906
mapping table 1311
restrict_process_product_association
AIM expanded listing rules 2296
AIM short listing rules 1907
mapping table 1401
restrict_product_categories_for_tool_part_relationship
AIM expanded listing rules 2296
AIM short listing rules 1908
restrict_product_concept_features_for_product_specification
AIM expanded listing rules 2297
AIM short listing rules 1909
mapping table 1372
restrict_version_assignment_for_action_directive
AIM expanded listing rules 2297
AIM short listing rules 1910
mapping table 1309
restrict_version_assignment_for_configurable_item
AIM expanded listing rules 2297
AIM short listing rules 1911
mapping table 1371
restrict_version_assignment_for_document
AIM expanded listing rules 2297
AIM short listing rules 1911

mapping table 667
restrict_version_assignment_for_draughting_model
AIM expanded listing rules 2297
AIM short listing rules 1912
mapping table 994, 1059
restrict_version_assignment_for_mechanical_design_geometric_presentation_representation
AIM expanded listing rules 2297
AIM short listing rules 1913
mapping table 994, 1059
restrict_version_assignment_for_physically_modelled_product_definition
AIM expanded listing rules 2298
AIM short listing rules 1913
mapping table 1050
restrict_version_assignment_for_presentation_area
AIM expanded listing rules 2298
AIM short listing rules 1914
mapping table 1001
restrict_version_assignment_for_process_plan
AIM expanded listing rules 2298
AIM short listing rules 1915
mapping table 1413
restrict_version_assignment_for_product_class
AIM expanded listing rules 2298
AIM short listing rules 1915
mapping table 1054, 1362
restrict_version_assignment_for_product_component
AIM expanded listing rules 2298
AIM short listing rules 1916
mapping table 1055, 1364
restrict_version_assignment_for_product_concept_feature
AIM expanded listing rules 2298
AIM short listing rules 1916
mapping table 1058, 1376
restrict_version_assignment_for_product_definition
AIM expanded listing rules 2298
AIM short listing rules 1917
mapping table 1056, 1368
restrict_version_assignment_for_representation
AIM expanded listing rules 2299
AIM short listing rules 1918
restrict_version_assignment_for_shape_representation
AIM expanded listing rules 2299
AIM short listing rules 1918
mapping table 864, 871, 885, 896, 911, 929, 955, 967, 1051
restrict_version_assignment_for_solution_based_product_definition
AIM expanded listing rules 2299
AIM short listing rules 1919
mapping table 1053, 1355
retention
AIM expanded listing entities 2157
AIM EXPRESS-G entity (figure H.38) 2457
AIM short listing entities 1681

mapping table 1244
retention_assignment
AIM expanded listing entities 2159
AIM EXPRESS-G entity (figure H.41) 2460
AIM short listing entities 1684
retention_item
AIM expanded listing types 1996
AIM EXPRESS-G types (figure H.45) 2464
AIM short listing types 1517
Retention_period
application assertion 486, 554–561
application object 321
ARM EXPRESS-G (figure G.12) 2379
mapping table 1244
retention_requires_retention_assignment
AIM expanded listing rules 2299
AIM short listing rules 1920
mapping table 1244
reversible_topology
AIM expanded listing types 1997
AIM EXPRESS-G types (figure H.84) 2503
reversible_topology_item
AIM expanded listing types 1997
AIM EXPRESS-G types (figure H.84) 2503
revolute_pair
AIM expanded listing entities 2159
AIM EXPRESS-G entity (figure H.116) 2535
revolute_pair_range
AIM expanded listing entities 2159
AIM EXPRESS-G entity (figure H.121) 2540
revolute_pair_value
AIM expanded listing entities 2159
AIM EXPRESS-G entity (figure H.114) 2533
revolved_face_solid
AIM expanded listing entities 2160
AIM EXPRESS-G entity (figure H.74) 2493
mapping table 958, 960
rib
AIM expanded listing entities 2160
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1685
mapping table 792, 811
Rib_feature
application assertion 561
application object 322
ARM EXPRESS-G (figure G.36) 2403
mapping table 811
Right_angular_wedge
application object 324
ARM EXPRESS-G (figure G.46) 2413
mapping table 958
right_angular_wedge

AIM expanded listing entities 2161
AIM EXPRESS-G entity (figure H.77) 2496
mapping table 945, 958
Right_circular_cone
application object 324
ARM EXPRESS-G (figure G.46) 2413
mapping table 958
right_circular_cone
AIM expanded listing entities 2161
AIM EXPRESS-G entity (figure H.77) 2496
mapping table 945, 958
Right_circular_cylinder
application object 325
ARM EXPRESS-G (figure G.46) 2413
mapping table 958
right_circular_cylinder
AIM expanded listing entities 2162
AIM EXPRESS-G entity (figure H.77) 2496
mapping table 945, 958
rigid_placement
AIM expanded listing types 1997
AIM EXPRESS-G types (figure H.116) 2535
rolling_curve_pair
AIM expanded listing entities 2162
AIM EXPRESS-G entity (figure H.117) 2536
rolling_curve_pair_value
AIM expanded listing entities 2162
AIM EXPRESS-G entity (figure H.114) 2533
rolling_surface_pair
AIM expanded listing entities 2162
AIM EXPRESS-G entity (figure H.117) 2536
rolling_surface_pair_value
AIM expanded listing entities 2162
AIM EXPRESS-G entity (figure H.114) 2533
rotation_about_direction
AIM expanded listing entities 2162
AIM EXPRESS-G entity (figure H.113) 2532
rotational_range_measure
AIM expanded listing types 1997
AIM EXPRESS-G types (figure H.120) 2539
Roughness
application assertion 561, 562
application object 325
ARM EXPRESS-G (figure G.50) 2417
mapping table 611
Round_hole
application assertion 562, 563
application object 327
ARM EXPRESS-G (figure G.36) 2403
mapping table 816
round_hole
AIM expanded listing entities 2163

AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1688
mapping table 792, 816
Rounded_u_profile
application assertion 563
application object 328
ARM EXPRESS-G (figure G.30) 2397
mapping table 822
rounded_u_profile
AIM expanded listing entities 2164
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1691
mapping table 778, 792, 822
runout_zone_definition
AIM expanded listing entities 2165
AIM EXPRESS-G entity (figure H.109) 2528
runout_zone_orientation
AIM expanded listing entities 2165
AIM EXPRESS-G entity (figure H.109) 2528
runout_zone_orientation_reference_direction
AIM expanded listing entities 2165
AIM EXPRESS-G entity (figure H.109) 2528

Same_time_machining_relationship
application assertion 563
application object 329
ARM EXPRESS-G (figure G.20) 2387
mapping table 1415
scalar_times_vector
AIM expanded listing functions 2242
screw_pair
AIM expanded listing entities 2165
AIM EXPRESS-G entity (figure H.117) 2536
screw_pair_range
AIM expanded listing entities 2165
AIM EXPRESS-G entity (figure H.122) 2541
screw_pair_value
AIM expanded listing entities 2165
AIM EXPRESS-G entity (figure H.114) 2533
seam_curve
AIM expanded listing entities 2166
AIM EXPRESS-G entity (figure H.66) 2485
second_in_minute
AIM expanded listing types 1997
AIM EXPRESS-G types (figure H.31) 2450
second_proj_axis
AIM expanded listing functions 2243
Security_classification
application assertion 473, 564–568
application object 330
ARM EXPRESS-G (figure G.3) 2370
mapping table 1149

security_classification
AIM expanded listing entities 2166
AIM EXPRESS-G entity (figure H.34) 2453
AIM short listing imported entity modifications 1762
mapping table 1149
security_classification_assignment
AIM expanded listing entities 2166
AIM EXPRESS-G entity (figure H.34) 2453
security_classification_item
AIM expanded listing types 1997
AIM EXPRESS-G types (figure H.34) 2453
AIM short listing types 1518
security_classification_level
AIM expanded listing entities 2166
AIM EXPRESS-G entity (figure H.34) 2453
AIM short listing imported entity modifications 1763
mapping table 1158
security_classification_requires_security_classification_assignment
AIM expanded listing rules 2299
AIM short listing rules 1920
mapping table 1149
security_classification_role
AIM expanded listing entities 2166
AIM EXPRESS-G entity (figure H.34) 2453
AIM short listing imported entity modifications 1763
Security_level
application assertion 407, 567, 568
application object 331
ARM EXPRESS-G (figure G.3) 2370
mapping table 1158
Selected_instance
application assertion 568
application object 331
ARM EXPRESS-G (figure G.14) 2381
mapping table 1204
Serial_configuration
application object 332
ARM EXPRESS-G (figure G.8) 2375
mapping table 1373
serial_numbered_effectivity
AIM expanded listing entities 2166
AIM EXPRESS-G entity (figure H.36) 2455
mapping table 1373, 1374
set_of_reversible_topology_item
AIM expanded listing types 1997
AIM EXPRESS-G types (figure H.84) 2503
set_of_topology_reversed
AIM expanded listing functions 2243
shading_curve_method
AIM expanded listing types 1997
AIM EXPRESS-G types (figure H.95) 2514
shading_surface_method

AIM expanded listing types 1998
AIM EXPRESS-G types (figure H.95) 2514
Shape_aspect
application assertion. 442, 456, 460, 501, 510, 518, 522, 542, 550, 568–570, 573, 580, 581, 584
application object 333
ARM EXPRESS-G (figure G.15) 2382
mapping table 1175
shape_aspect
AIM expanded listing entities 2166
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing imported entity modifications 1763
mapping table 602, 672, 674, 676, 682, 699–701, 745, 798, 799, 801, 823, 824, 838, 849, 1175, 1426, 1429, 1435, 1446, 1448
shape_aspect_associativity
AIM expanded listing entities 2167
AIM EXPRESS-G entity (figure H.12) 2431
shape_aspect_deriving_relationship
AIM expanded listing entities 2167
AIM EXPRESS-G entity (figure H.12) 2431
Shape_aspect_relationship
application assertion 428, 510, 520, 523, 569, 570
application object 333
ARM EXPRESS-G (figure G.15) 2382
mapping table 1176
shape_aspect_relationship
AIM expanded listing entities 2167
AIM EXPRESS-G entity (figure H.12) 2431
AIM short listing imported entity modifications 1763
mapping table 603, 717, 723, 877, 881, 903, 906, 921, 924, 946, 949, 1176, 1290, 1292, 1426–1428, 1435, 1447
shape_aspect_relationship_subtype_exclusiveness
AIM expanded listing rules 2299
AIM short listing rules 1921
shape_aspect_subtype_exclusiveness
AIM expanded listing rules 2300
AIM short listing rules 1922
shape_aspect_transition
AIM expanded listing entities 2167
AIM EXPRESS-G entity (figure H.12) 2431
AIM short listing entities 1693
mapping table 877, 880, 903, 906, 921, 924, 946, 949
shape_defining_relationship
AIM expanded listing entities 2168
AIM EXPRESS-G entity (figure H.12) 2431
AIM short listing entities 1695
mapping table 709
shape_definition
AIM expanded listing types 1998
AIM EXPRESS-G types (figure H.10) 2429
shape_definition_representation
AIM expanded listing entities 2168
AIM EXPRESS-G entity (figure H.13) 2432

mapping table 1176
Shape_dependent_property
application assertion 569
application object 335
ARM EXPRESS-G (figure G.24) 2391
mapping table 1043
Shape_description_association
application assertion 569, 570
application object 335
ARM EXPRESS-G (figure G.15) 2382
mapping table 1176
shape_dimension_representation
AIM expanded listing entities 2168
AIM EXPRESS-G entity (figure H.20) 2439
AIM short listing imported entity modifications 1764
mapping table 1444
shape_representation
AIM expanded listing entities 2169
AIM EXPRESS-G entity (figure H.20) 2439
AIM short listing imported entity modifications 1764
mapping table 864, 871, 885, 927, 967
shape_representation_relationship
AIM expanded listing entities 2169
AIM EXPRESS-G entity (figure H.13) 2432
shape_representation_subtype_exclusiveness
AIM expanded listing rules 2301
AIM short listing rules 1924
mapping table 862, 869, 883, 894, 909, 927, 952, 965
shape_representation_with_parameters
AIM expanded listing entities 2169
AIM EXPRESS-G entity (figure H.20) 2439
AIM short listing entities 1695
sheets_belong_to_one_drawing
AIM expanded listing rules 2301
AIM short listing rules 1925
mapping table 632
Shell
application assertion 491, 492, 498, 571, 580
application object 336
ARM EXPRESS-G (figure G.44) 2411
mapping table 888
shell
AIM expanded listing types 1998
AIM EXPRESS-G types (figure H.78) 2497
shell_based_surface_model
AIM expanded listing entities 2169
AIM EXPRESS-G entity (figure H.78) 2497
shell_reversed
AIM expanded listing functions 2244
si_prefix
AIM expanded listing types 1998
AIM EXPRESS-G types (figure H.47) 2466

si_unit
AIM expanded listing entities 2169
AIM EXPRESS-G entity (figure H.47) 2466
si_unit_name
AIM expanded listing types 1998
AIM EXPRESS-G types (figure H.47) 2466
Significant_number_of_digits
application assertion 584
application object 336
ARM EXPRESS-G (figure G.26) 2393
mapping table 1440
Simple_pair_range
application assertion 571
application object 336
ARM EXPRESS-G (figure G.22) 2389
mapping table 977
simple_pair_range
AIM expanded listing entities 2169
AIM EXPRESS-G entity (figure H.118) 2537
mapping table 977
Simulation_information
application assertion 403, 415, 447, 454, 473, 480, 536, 559, 567, 571–573, 597
application object 336
ARM EXPRESS-G (figure G.21) 2388
mapping table 977
Simultaneous_activity
application assertion 573
application object 337
ARM EXPRESS-G (figure G.17) 2384
mapping table 1308
Simultaneous_process_operation
application assertion 573
application object 338
ARM EXPRESS-G (figure G.19) 2386
mapping table 1419
Single_instance
application assertion 505
application object 338
ARM EXPRESS-G (figure G.14) 2381
mapping table 1206
Size_dimension
application assertion 573
application object 339
ARM EXPRESS-G (figure G.47) 2414
mapping table 1440
size_select
AIM expanded listing types 1998
AIM EXPRESS-G types (figure H.100) 2519
sliding_curve_pair
AIM expanded listing entities 2169
AIM EXPRESS-G entity (figure H.117) 2536
sliding_curve_pair_value

AIM expanded listing entities 2170
AIM EXPRESS-G entity (figure H.115) 2534
sliding_surface_pair
AIM expanded listing entities 2170
AIM EXPRESS-G entity (figure H.117) 2536
sliding_surface_pair_value
AIM expanded listing entities 2170
AIM EXPRESS-G entity (figure H.115) 2534
slot
AIM expanded listing entities 2170
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1696
mapping table 792, 824
slot_end
AIM expanded listing entities 2172
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1699
mapping table 746, 778, 779, 794, 795, 823, 846
Slot_end_type
application assertion 574
application object 339
ARM EXPRESS-G (figure G.36) 2403
mapping table 823
Slot_feature
application assertion 574
application object 339
ARM EXPRESS-G (figure G.36) 2403
mapping table 824
solid_angle_measure
AIM expanded listing types 1998
AIM EXPRESS-G types (figure H.48) 2467
solid_angle_measure_with_unit
AIM expanded listing entities 2173
AIM EXPRESS-G entity (figure H.46) 2465
mapping table 1434
solid_angle_unit
AIM expanded listing entities 2173
AIM EXPRESS-G entity (figure H.47) 2466
Solid_fill_area
application assertion 409
application object 341
ARM EXPRESS-G (figure G.31) 2398
mapping table 1009
solid_model
AIM expanded listing entities 2173
AIM EXPRESS-G entity (figure H.74) 2493
Solid_of_linear_extrusion
application object 341
ARM EXPRESS-G (figure G.46) 2413
mapping table 958
Solid_of_revolution
application object 341

ARM EXPRESS-G (figure G.46) 2413
mapping table 958
Solid_replica
application assertion 423, 441, 574, 575
application object 341
ARM EXPRESS-G (figure G.46) 2413
mapping table 958
solid_replica
AIM expanded listing entities 2173
AIM EXPRESS-G entity (figure H.75) 2494
solution_alternative_definition
AIM expanded listing entities 2174
AIM short listing entities 1701
solution_based_product_definition
AIM expanded listing entities 2174
AIM EXPRESS-G entity (figure H.4) 2423
AIM short listing entities 1702
mapping table 1350, 1354, 1381, 1382
Solution_certification
application assertion 575
application object 342
ARM EXPRESS-G (figure G.8) 2375
mapping table 1374
Solution_instance_association
application assertion 575
application object 343
ARM EXPRESS-G (figure G.8) 2375
mapping table 1376
source_item
AIM expanded listing types 1998
AIM EXPRESS-G types (figure H.50) 2469
spatial_rotation
AIM expanded listing types 1998
AIM EXPRESS-G types (figure H.113) 2532
Specific_classification
application object 343
ARM EXPRESS-G (figure G.1) 2368
mapping table 1158, 1336
Specification
application assertion 403, 407, 415, 430, 447, 454, 467, 473, 480, 486, 536, 548, 560, 567, 576, 577, 590, 597
application object 345
ARM EXPRESS-G (figure G.6) 2373
mapping table 1376
Specification_category
application assertion 404, 407, 415, 429, 448, 454, 467, 473, 480, 486, 536, 560, 567, 576, 590, 597
application object 346
ARM EXPRESS-G (figure G.6) 2373
mapping table 1378
Specification_expression
application assertion 404, 415, 429, 448, 454, 473, 480, 536, 560, 576, 577, 597

application object 347
ARM EXPRESS-G (figure G.6) 2373
mapping table 1378
Specification_inclusion
application assertion 404, 416, 430, 448, 455, 473, 480, 536, 560, 577, 597
application object 348
ARM EXPRESS-G (figure G.6) 2373
mapping table 1380
Specified_higher_usage
application assertion 577
application object 349
ARM EXPRESS-G (figure G.13) 2380
mapping table 1381
specified_higher_usage_occurrence
AIM expanded listing entities 2175
AIM EXPRESS-G entity (figure H.6) 2425
mapping table 1381
specified_item
AIM expanded listing types 1998
AIM EXPRESS-G types (figure H.124) 2543
AIM short listing types 1519
Sphere
application object 350
ARM EXPRESS-G (figure G.46) 2413
mapping table 960
sphere
AIM expanded listing entities 2175
AIM EXPRESS-G entity (figure H.76) 2495
mapping table 945
Spherical_bottom
application assertion 577
application object 350
ARM EXPRESS-G (figure G.36) 2403
mapping table 829
spherical_pair
AIM expanded listing entities 2175
AIM EXPRESS-G entity (figure H.116) 2535
spherical_pair_range
AIM expanded listing entities 2175
AIM EXPRESS-G entity (figure H.122) 2541
spherical_pair_value
AIM expanded listing entities 2176
AIM EXPRESS-G entity (figure H.115) 2534
Spherical_surface
application object 350
ARM EXPRESS-G (figure G.44) 2411
mapping table 888, 914, 932, 960, 968
spherical_surface
AIM expanded listing entities 2176
AIM EXPRESS-G entity (figure H.70) 2489
mapping table 888, 914, 932, 960, 968
Square_u_profile

application assertion 552, 578
application object 351
ARM EXPRESS-G (figure G.30) 2397
mapping table 829
square_u_profile
AIM expanded listing entities 2176
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1704
mapping table 778, 792, 829, 830
standard_uncertainty
AIM expanded listing entities 2177
AIM EXPRESS-G entity (figure H.56) 2475
mapping table 1439–1441
Straightness_tolerance
application assertion 578
application object 352
ARM EXPRESS-G (figure G.48) 2415
mapping table 1459
straightness_tolerance
AIM expanded listing entities 2177
AIM EXPRESS-G entity (figure H.109) 2528
mapping table 1459
String_with_language
application assertion 526
application object 353
ARM EXPRESS-G (figure G.4) 2371
mapping table 1159
structured_dimension_callout
AIM expanded listing entities 2177
AIM EXPRESS-G entity (figure H.106) 2525
mapping table 624, 626, 648–650
Structuring_method_hierarchy
application assertion 578
application object 354
ARM EXPRESS-G (figure G.41) 2408
mapping table 1179
style_context_select
AIM expanded listing types 1998
AIM EXPRESS-G types (figure H.93) 2512
styled_curve
AIM expanded listing rules 2301
AIM short listing rules 1926
mapping table 993
Styled_element
application assertion 578, 579
application object 354
ARM EXPRESS-G (figure G.35) 2402
mapping table 991
Styled_geometric_model
application assertion 579
application object 355
ARM EXPRESS-G (figure G.25) 2392

mapping table 991
styled_item
AIM expanded listing entities 2178
AIM EXPRESS-G entity (figure H.93) 2512
AIM short listing imported entity modifications 1764
mapping table 993
Styled_model
application assertion 404, 408, 416, 448, 455, 474, 480, 496, 497, 525, 536, 560, 567, 579, 582, 598
application object 355
ARM EXPRESS-G (figure G.25) 2392
mapping table 991
Styled_model_element
application assertion 579
application object 356
ARM EXPRESS-G (figure G.35) 2402
mapping table 993
Styled_model_relationship
application assertion 525, 579
application object 356
ARM EXPRESS-G (figure G.25) 2392
mapping table 993
Styled_model_version
application object 357
ARM EXPRESS-G (figure G.25) 2392
mapping table 994
subtype_mandatory_camera_image_2d_or_3d
AIM expanded listing rules 2302
AIM short listing rules 1927
mapping table 987
subtype_mandatory_camera_model_2d_or_3d
AIM expanded listing rules 2302
AIM short listing rules 1927
mapping table 987
subtype_mandatory_drawing_revision
AIM expanded listing rules 2302
AIM short listing rules 1928
mapping table 626
subtype_mandatory_effectivity
AIM expanded listing rules 2302
AIM short listing rules 1928
mapping table 1209
subtype_mandatory_pre_defined_colour
AIM expanded listing rules 2302
AIM short listing rules 1930
mapping table 987, 990, 1015, 1016
subtype_mandatory_pre_defined_curve_font
AIM expanded listing rules 2303
AIM short listing rules 1930
mapping table 997, 1015
subtype_mandatory_pre_defined_text_font
AIM expanded listing rules 2303

AIM short listing rules 1931
mapping table 1009
subtype_mandatory_product_concept
AIM expanded listing rules 2303
AIM short listing rules 1931
subtype_mandatory_product_definition_with_associated_documents
AIM expanded listing rules 2303
AIM short listing rules 1932
mapping table 669
suitably_based_mechanism
AIM expanded listing functions 2244
Supplier_solution
application object 357
ARM EXPRESS-G (figure G.8) 2375
mapping table 1381
supported_item
AIM expanded listing types 1999
AIM EXPRESS-G types (figure H.40) 2459
Surface
application assertion 438, 487, 491, 492, 498, 500
application object 358
ARM EXPRESS-G (figure G.44) 2411
mapping table 888, 914, 932, 960, 968
surface
AIM expanded listing entities 2178
AIM EXPRESS-G entity (figure H.69) 2488
mapping table 873, 890, 899, 916, 932, 939
Surface_appearance
application assertion 409, 580
application object 358
ARM EXPRESS-G (figure G.31) 2398
mapping table 995, 1017
Surface_appearance_element
application assertion 580
application object 359
ARM EXPRESS-G (figure G.31) 2398
mapping table 995, 1018
Surface_condition
application assertion 486, 580
application object 359
ARM EXPRESS-G (figure G.50) 2417
mapping table 612
surface_condition
AIM expanded listing entities 2178
AIM EXPRESS-G entity (figure H.10) 2429
AIM short listing entities 1707
surface_condition_callout
AIM expanded listing entities 2185
AIM EXPRESS-G entity (figure H.106) 2525
AIM short listing entities 1719
mapping table 649
Surface_condition_symbol

application object 360
ARM EXPRESS-G (figure G.33) 2400
mapping table 649
surface_curve
AIM expanded listing entities 2185
AIM EXPRESS-G entity (figure H.66) 2485
mapping table 877, 903, 920, 945, 965
Surface_model
application assertion 580
application object 360
ARM EXPRESS-G (figure G.44) 2411
mapping table 888
surface_model
AIM expanded listing types 1999
AIM EXPRESS-G types (figure H.124) 2543
Surface_of_linear_extrusion
application object 360
ARM EXPRESS-G (figure G.44) 2411
mapping table 889, 914, 932, 960, 968
surface_of_linear_extrusion
AIM expanded listing entities 2185
AIM EXPRESS-G entity (figure H.71) 2490
mapping table 889, 914, 932, 960, 968
Surface_of_revolution
application object 361
ARM EXPRESS-G (figure G.44) 2411
mapping table 889, 914, 932, 960, 968
surface_of_revolution
AIM expanded listing entities 2185
AIM EXPRESS-G entity (figure H.71) 2490
mapping table 889, 914, 932, 960, 968
surface_pair
AIM expanded listing entities 2185
AIM EXPRESS-G entity (figure H.117) 2536
surface_pair_range
AIM expanded listing entities 2186
AIM EXPRESS-G entity (figure H.122) 2541
surface_patch
AIM expanded listing entities 2186
AIM EXPRESS-G entity (figure H.73) 2492
Surface_profile_tolerance
application assertion 580
application object 361
ARM EXPRESS-G (figure G.48) 2415
mapping table 1460
surface_profile_tolerance
AIM expanded listing entities 2186
AIM EXPRESS-G entity (figure H.109) 2528
mapping table 1460
surface_rendering_properties
AIM expanded listing entities 2187
AIM EXPRESS-G entity (figure H.95) 2514

surface_replica
AIM expanded listing entities 2187
AIM EXPRESS-G entity (figure H.71) 2490
surface_side
AIM expanded listing types 1999
AIM EXPRESS-G types (figure H.94) 2513
surface_side_style
AIM expanded listing entities 2187
AIM EXPRESS-G entity (figure H.94) 2513
surface_side_style_select
AIM expanded listing types 1999
AIM EXPRESS-G types (figure H.94) 2513
surface_style_boundary
AIM expanded listing entities 2187
AIM EXPRESS-G entity (figure H.95) 2514
mapping table 995, 1018
surface_style_control_grid
AIM expanded listing entities 2187
AIM EXPRESS-G entity (figure H.95) 2514
mapping table 1018, 1022
surface_style_element_select
AIM expanded listing types 1999
AIM EXPRESS-G types (figure H.95) 2514
surface_style_fill_area
AIM expanded listing entities 2187
AIM EXPRESS-G entity (figure H.95) 2514
mapping table 995, 1001, 1018
surface_style_parameter_line
AIM expanded listing entities 2187
AIM EXPRESS-G entity (figure H.95) 2514
mapping table 995, 997, 1018, 1021
surface_style_reflectance_ambient
AIM expanded listing entities 2187
AIM EXPRESS-G entity (figure H.95) 2514
surface_style_reflectance_ambient_diffuse
AIM expanded listing entities 2187
surface_style_reflectance_ambient_diffuse_specular
AIM expanded listing entities 2188
surface_style_rendering
AIM expanded listing entities 2188
AIM EXPRESS-G entity (figure H.95) 2514
mapping table 1019
surface_style_rendering_with_properties
AIM expanded listing entities 2188
AIM EXPRESS-G entity (figure H.95) 2514
mapping table 1019
surface_style_segmentation_curve
AIM expanded listing entities 2188
AIM EXPRESS-G entity (figure H.95) 2514
mapping table 1018
surface_style_silhouette
AIM expanded listing entities 2188

AIM EXPRESS-G entity (figure H.95) 2514
mapping table 1018
surface_style_transparent
AIM expanded listing entities 2188
AIM EXPRESS-G entity (figure H.95) 2514
surface_style_usage
AIM expanded listing entities 2188
AIM EXPRESS-G entity (figure H.94) 2513
mapping table 995, 1017
surface_weights_positive
AIM expanded listing functions 2245
Swept_face_solid
application assertion 424, 441, 575, 581
application object 361
ARM EXPRESS-G (figure G.46) 2413
mapping table 960
swept_face_solid
AIM expanded listing entities 2188
AIM EXPRESS-G entity (figure H.74) 2493
swept_surface
AIM expanded listing entities 2189
AIM EXPRESS-G entity (figure H.71) 2490
symbol_colour
AIM expanded listing entities 2189
AIM EXPRESS-G entity (figure H.98) 2517
AIM short listing imported entity modifications 1764
symbol_representation
AIM expanded listing entities 2189
AIM EXPRESS-G entity (figure H.86) 2505
symbol_representation_map
AIM expanded listing entities 2189
AIM EXPRESS-G entity (figure H.17) 2436
symbol_style
AIM expanded listing entities 2189
AIM EXPRESS-G entity (figure H.98) 2517
symbol_style_select
AIM expanded listing types 1999
AIM EXPRESS-G types (figure H.98) 2517
symbol_target
AIM expanded listing entities 2189
AIM EXPRESS-G entity (figure H.88) 2507
symmetric_shape_aspect
AIM expanded listing entities 2189
AIM EXPRESS-G entity (figure H.11) 2430
Symmetry_tolerance
application assertion 581
application object 361
ARM EXPRESS-G (figure G.48) 2415
mapping table 1460
symmetry_tolerance
AIM expanded listing entities 2190
AIM EXPRESS-G entity (figure H.110) 2529

mapping table 1460

Tactile_appearance
application assertion 581
application object 362
ARM EXPRESS-G (figure G.50) 2417
mapping table 615
tangent
AIM expanded listing entities 2190
AIM EXPRESS-G entity (figure H.54) 2473
taper
AIM expanded listing entities 2190
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1719
mapping table 681, 741, 742
Target_area
application assertion 581
application object 363
ARM EXPRESS-G (figure G.49) 2416
mapping table 1461
Target_circle
application object 363
ARM EXPRESS-G (figure G.49) 2416
mapping table 1461
Target_line
application object 364
ARM EXPRESS-G (figure G.49) 2416
mapping table 1462
Target_point
application object 364
ARM EXPRESS-G (figure G.49) 2416
mapping table 1463
Target_rectangle
application object 364
ARM EXPRESS-G (figure G.49) 2416
mapping table 1463
Technical_solution
application assertion 581
application object 364
ARM EXPRESS-G (figure G.8) 2375
mapping table 1382
Tee_profile
application assertion 582
application object 365
ARM EXPRESS-G (figure G.30) 2397
mapping table 832
tee_profile
AIM expanded listing entities 2191
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1721
mapping table 778, 792, 832, 833
Template_instance

application assertion 418, 498, 502, 506, 520, 563, 579, 582, 583, 585
application object 367
ARM EXPRESS-G (figure G.41) 2408
mapping table 1180
terminator_symbol
AIM expanded listing entities 2193
AIM EXPRESS-G entity (figure H.86) 2505
AIM short listing imported entity modifications 1765
terminator_symbol_constraint
AIM expanded listing rules 2303
AIM short listing rules 1933
mapping table 1008
Text
application assertion 419–421
application object 368
ARM EXPRESS-G (figure G.34) 2401
mapping table 1009
text
AIM expanded listing types 1999
text_alignment
AIM expanded listing types 1999
AIM EXPRESS-G types (figure H.89) 2508
Text_appearance
application assertion 409
application object 368
ARM EXPRESS-G (figure G.31) 2398
mapping table 1010
text_delineation
AIM expanded listing types 1999
AIM EXPRESS-G types (figure H.89) 2508
text_font_usage
AIM expanded listing rules 2303
AIM short listing rules 1933
mapping table 1009
text_literal
AIM expanded listing entities 2193
AIM EXPRESS-G entity (figure H.89) 2508
text_literal_with_associated_curves
AIM expanded listing entities 2193
AIM EXPRESS-G entity (figure H.89) 2508
text_literal_with_blanking_box
AIM expanded listing entities 2193
AIM EXPRESS-G entity (figure H.89) 2508
text_literal_with_delineation
AIM expanded listing entities 2193
AIM EXPRESS-G entity (figure H.89) 2508
text_literal_with_extent
AIM expanded listing entities 2193
AIM EXPRESS-G entity (figure H.89) 2508
text_or_character
AIM expanded listing types 1999
AIM EXPRESS-G types (figure H.87) 2506

text_path
AIM expanded listing types 1999
AIM EXPRESS-G types (figure H.89) 2508
text_string_representation
AIM expanded listing entities 2194
text_style
AIM expanded listing entities 2194
AIM EXPRESS-G entity (figure H.100) 2519
AIM short listing imported entity modifications 1765
text_style_for_defined_font
AIM expanded listing entities 2194
AIM EXPRESS-G entity (figure H.100) 2519
AIM short listing imported entity modifications 1765
text_style_with_box_characteristics
AIM expanded listing entities 2194
AIM EXPRESS-G entity (figure H.100) 2519
mapping table 1010, 1012
text_style_with_mirror
AIM expanded listing entities 2194
AIM EXPRESS-G entity (figure H.100) 2519
mapping table 1010
text_style_with_spacing
AIM expanded listing entities 2195
AIM EXPRESS-G entity (figure H.100) 2519
thermodynamic_temperature_measure
AIM expanded listing types 2000
AIM EXPRESS-G types (figure H.48) 2467
thermodynamic_temperature_measure_with_unit
AIM expanded listing entities 2195
AIM EXPRESS-G entity (figure H.46) 2465
thermodynamic_temperature_unit
AIM expanded listing entities 2195
AIM EXPRESS-G entity (figure H.47) 2466
Thickness
application assertion 583
application object 368
ARM EXPRESS-G (figure G.24) 2391
mapping table 1043
thread
AIM expanded listing entities 2195
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1726
mapping table 735, 840
Thread_feature
application assertion 435, 494, 505, 511, 523, 570, 583
application object 370
ARM EXPRESS-G (figure G.39) 2406
mapping table 838
thread_occurrence
AIM expanded listing entities 2197
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1729

mapping table 717, 718, 838
Thread_specification
application assertion 583
application object 371
ARM EXPRESS-G (figure G.39) 2406
mapping table 840
Through_bottom_condition
application object 371
ARM EXPRESS-G (figure G.37) 2404
mapping table 841
Through_hole_condition
application object 371
ARM EXPRESS-G (figure G.36) 2403
mapping table 841
Tiling
application assertion 410
application object 371
ARM EXPRESS-G (figure G.31) 2398
mapping table 1010
time_measure
AIM expanded listing types 2000
AIM EXPRESS-G types (figure H.48) 2467
time_measure_with_unit
AIM expanded listing entities 2198
AIM EXPRESS-G entity (figure H.46) 2465
time_unit
AIM expanded listing entities 2198
AIM EXPRESS-G entity (figure H.47) 2466
tolerance_method_definition
AIM expanded listing types 2000
AIM EXPRESS-G types (figure H.110) 2529
tolerance_select
AIM expanded listing types 2000
AIM EXPRESS-G types (figure H.124) 2543
Tolerance_table_cell
application assertion 583, 584
application object 372
ARM EXPRESS-G (figure G.26) 2393
mapping table 1441
tolerance_value
AIM expanded listing entities 2198
AIM EXPRESS-G entity (figure H.110) 2529
AIM short listing imported entity modifications 1765
mapping table 1439
Tolerance_zone
application assertion 550, 584
application object 373
ARM EXPRESS-G (figure G.49) 2416
mapping table 1465
tolerance_zone
AIM expanded listing entities 2198
AIM EXPRESS-G entity (figure H.109) 2528

Tolerance_zone_definition
application assertion ... 584
application object ... 373
ARM EXPRESS-G (figure G.49) ... 2416
mapping table ... 1465
tolerance_zone_definition
AIM expanded listing entities ... 2198
AIM EXPRESS-G entity (figure H.109) ... 2528
mapping table ... 1465
tolerance_zone_form
AIM expanded listing entities ... 2198
AIM EXPRESS-G entity (figure H.109) ... 2528
mapping table ... 1465
Tool_attachment_point_frame
application object ... 374
ARM EXPRESS-G (figure G.21) ... 2388
mapping table ... 981
Tool_center_point_frame
application object ... 374
ARM EXPRESS-G (figure G.21) ... 2388
mapping table ... 981
Tool_part_relationship
application assertion ... 585
application object ... 374
ARM EXPRESS-G (figure G.20) ... 2387
mapping table ... 1420
topological_representation_item
AIM expanded listing entities ... 2199
AIM EXPRESS-G entity (figure H.80) ... 2499
Topologically_connected_set
application assertion ... 435, 585
application object ... 375
ARM EXPRESS-G (figure G.45) ... 2412
mapping table ... 932
topology_reversed
AIM expanded listing functions ... 2246
Toroidal_surface
application object ... 376
ARM EXPRESS-G (figure G.44) ... 2411
mapping table ... 889, 914, 933, 961, 969
toroidal_surface
AIM expanded listing entities ... 2199
AIM EXPRESS-G entity (figure H.70) ... 2489
mapping table ... 889, 933, 969
Torus
application object ... 376
ARM EXPRESS-G (figure G.46) ... 2413
mapping table ... 961
torus
AIM expanded listing entities ... 2199
AIM EXPRESS-G entity (figure H.76) ... 2495
mapping table ... 945

Total_runout_tolerance
application assertion 585
application object 376
ARM EXPRESS-G (figure G.48) 2415
mapping table 1466
total_runout_tolerance
AIM expanded listing entities 2199
AIM EXPRESS-G entity (figure H.110) 2529
mapping table 1466
Transformation
application assertion 422, 425, 432, 434, 435, 489, 499, 539, 542, 544, 554, 561, 563, 574, 583
application object 377
ARM EXPRESS-G (figure G.33) 2400
mapping table 1181
transformation
AIM expanded listing types 2000
AIM EXPRESS-G types (figure H.18) 2437
Transformation_2d
application assertion 587
application object 377
ARM EXPRESS-G (figure G.33) 2400
mapping table 1181
Transformation_3d
application object 377
ARM EXPRESS-G (figure G.33) 2400
mapping table 1182
Transformation_to_reference_frame
application assertion 515, 516
application object 377
ARM EXPRESS-G (figure G.21) 2388
mapping table 981
transition_code
AIM expanded listing types 2000
AIM EXPRESS-G types (figure H.65) 2484
Transition_feature
application assertion 434, 435, 494, 505, 511, 523, 570, 585
application object 377
ARM EXPRESS-G (figure G.29) 2396
mapping table 842
transition_feature
AIM expanded listing entities 2199
AIM EXPRESS-G entity (figure H.108) 2527
AIM short listing entities 1731
mapping table 712, 717, 842
translational_range_measure
AIM expanded listing types 2000
AIM EXPRESS-G types (figure H.120) 2539
Treatment_result
application assertion 586
application object 378
ARM EXPRESS-G (figure G.50) 2417
mapping table 616

Trimmed_curve
application object 379
ARM EXPRESS-G (figure G.43) 2410
mapping table 865, 872, 969
trimmed_curve
AIM expanded listing entities 2200
AIM EXPRESS-G entity (figure H.63) 2482
Trimmed_surface
application object 379
ARM EXPRESS-G (figure G.44) 2411
mapping table 969
trimming_preference
AIM expanded listing types 2000
AIM EXPRESS-G types (figure H.63) 2482
trimming_select
AIM expanded listing types 2000
AIM EXPRESS-G types (figure H.63) 2482
two_direction_repeat_factor
AIM expanded listing entities 2200
AIM EXPRESS-G entity (figure H.98) 2517
type_qualifier
AIM expanded listing entities 2200
AIM EXPRESS-G entity (figure H.56) 2475
mapping table 1444

u_direction_count
AIM expanded listing types 2000
AIM EXPRESS-G types (figure H.95) 2514
uncertainty_assigned_representation
AIM expanded listing entities 2200
AIM EXPRESS-G entity (figure H.16) 2435
mapping table 1161
uncertainty_measure_with_unit
AIM expanded listing entities 2200
AIM EXPRESS-G entity (figure H.19) 2438
mapping table 862, 869, 883, 894, 909, 927, 953, 965, 1161, 1162
uncertainty_qualifier
AIM expanded listing entities 2200
AIM EXPRESS-G entity (figure H.56) 2475
AIM short listing imported entity modifications 1765
unconstrained_pair
AIM expanded listing entities 2201
AIM EXPRESS-G entity (figure H.116) 2535
unconstrained_pair_value
AIM expanded listing entities 2201
AIM EXPRESS-G entity (figure H.115) 2534
uniform_curve
AIM expanded listing entities 2201
AIM EXPRESS-G entity (figure H.64) 2483
uniform_surface
AIM expanded listing entities 2201
AIM EXPRESS-G entity (figure H.72) 2491

unique_link_usage
AIM expanded listing functions 2246
unit
AIM expanded listing types 2000
AIM EXPRESS-G types (figure H.46) 2465
universal_pair
AIM expanded listing entities 2201
AIM EXPRESS-G entity (figure H.117) 2536
universal_pair_range
AIM expanded listing entities 2201
AIM EXPRESS-G entity (figure H.122) 2541
universal_pair_value
AIM expanded listing entities 2202
AIM EXPRESS-G entity (figure H.112) 2531
unlimited_range
AIM expanded listing types 2001
Upper_lower_limit
application assertion 584
application object 379
ARM EXPRESS-G (figure G.26) 2393
mapping table 1443
Upper_lower_toleranced_datum
application assertion 584
application object 380
ARM EXPRESS-G (figure G.26) 2393
mapping table 1443
using_representations
AIM expanded listing functions 2247
AIM short listing functions 1939

v_direction_count
AIM expanded listing types 2001
AIM EXPRESS-G types (figure H.95) 2514
valid_basis_curve_in_2d_wireframe
AIM expanded listing functions 2247
valid_calendar_date
AIM expanded listing functions 2248
valid_geometrically_bounded_wf_curve
AIM expanded listing functions 2248
valid_geometrically_bounded_wf_point
AIM expanded listing functions 2249
valid_time
AIM expanded listing functions 2249
valid_units
AIM expanded listing functions 2250
valid_wireframe_edge_curve
AIM expanded listing functions 2252
valid_wireframe_vertex_point
AIM expanded listing functions 2252
Value_limit
application assertion 463
application object 380

ARM EXPRESS-G (figure G.47) 2414
mapping table 1443
value_qualifier
AIM expanded listing types 2001
AIM EXPRESS-G types (figure H.56) 2475
Value_range
application assertion 464
application object 380
ARM EXPRESS-G (figure G.47) 2414
mapping table 1444
Value_termination
application assertion 586
application object 381
ARM EXPRESS-G (figure G.38) 2405
mapping table 843
Value_with_unit
application assertion 398, 432, 457, 459, 474, 475, 480, 489, 503, 526, 550–552, 560, 562, 568, 581
application object 382
ARM EXPRESS-G (figure G.23) 2390
mapping table 618, 677, 1044, 1207, 1266, 1309
vector
AIM expanded listing entities 2202
AIM EXPRESS-G entity (figure H.59) 2478
Vector_appearance
application assertion 410
application object 382
ARM EXPRESS-G (figure G.31) 2398
mapping table 997
vector_difference
AIM expanded listing functions 2253
vector_or_direction
AIM expanded listing types 2001
AIM EXPRESS-G types (figure H.68) 2487
vector_style
AIM expanded listing entities 2202
AIM EXPRESS-G entity (figure H.94) 2513
AIM short listing entities 1732
mapping table 997
Vee_profile
application assertion 586
application object 382
ARM EXPRESS-G (figure G.30) 2397
mapping table 844
vee_profile
AIM expanded listing entities 2202
AIM EXPRESS-G entity (figure H.11) 2430
AIM short listing entities 1732
mapping table 778, 792, 844
version_assignment
AIM expanded listing entities 2203
AIM EXPRESS-G entity (figure H.9) 2428

AIM short listing entities 1735
version_item
AIM expanded listing types 2001
AIM EXPRESS-G types (figure H.9) 2428
AIM short listing types 1520
versioned_action_request
AIM expanded listing entities 2203
AIM EXPRESS-G entity (figure H.39) 2458
AIM short listing imported entity modifications 1766
mapping table 1310, 1311
versioned_action_request_requires_date_and_person_or_organization
AIM expanded listing rules 2304
AIM short listing rules 1934
mapping table 1311
vertex
AIM expanded listing entities 2203
AIM EXPRESS-G entity (figure H.80) 2499
Vertex_loop
application assertion 487, 586
application object 383
ARM EXPRESS-G (figure G.44) 2411
mapping table 889, 914, 933, 961
vertex_loop
AIM expanded listing entities 2203
AIM EXPRESS-G entity (figure H.81) 2500
mapping table 933
Vertex_point
application assertion 397, 438, 468, 528, 586
application object 383
ARM EXPRESS-G (figure G.44) 2411
mapping table 889, 914, 933, 961
vertex_point
AIM expanded listing entities 2203
AIM EXPRESS-G entity (figure H.80) 2499
mapping table 873, 899, 916, 933, 939
View
application assertion 440, 525, 587, 591
application object 384
ARM EXPRESS-G (figure G.32) 2399
mapping table 997
View_area
application assertion 404, 416, 417, 440, 448, 455, 459, 474, 481, 486, 537, 561, 568, 587, 590, 598
application object 386
ARM EXPRESS-G (figure G.32) 2399
mapping table 1001
View_area_assignment
application assertion 588–590
application object 387
ARM EXPRESS-G (figure G.32) 2399
mapping table 633
View_area_relationship

application assertion 525, 590
application object 387
ARM EXPRESS-G (figure G.32) 2399
mapping table 1010
View_placed_annotation
application assertion 590, 591
application object 388
ARM EXPRESS-G (figure G.32) 2399
mapping table 1012
View_relationship
application assertion 591
application object 389
ARM EXPRESS-G (figure G.32) 2399
mapping table 1013
view_volume
AIM expanded listing entities 2204
AIM EXPRESS-G entity (figure H.92) 2511
mapping table 987, 988
Visual_appearance
application assertion 580, 591
application object 390
ARM EXPRESS-G (figure G.50) 2417
mapping table 618
volume_measure
AIM expanded listing types 2001
AIM EXPRESS-G types (figure H.48) 2467
volume_measure_with_unit
AIM expanded listing entities 2204
AIM EXPRESS-G entity (figure H.46) 2465
volume_unit
AIM expanded listing entities 2204
AIM EXPRESS-G entity (figure H.47) 2466

wireframe_model
AIM expanded listing types 2001
AIM EXPRESS-G types (figure H.124) 2543
Wireframe_model_2d
application assertion 591
application object 392
ARM EXPRESS-G (figure G.43) 2410
mapping table 865
Wireframe_model_3d
application assertion 591
application object 392
ARM EXPRESS-G (figure G.43) 2410
mapping table 872
Woodruff_slot_end_type
application assertion 592
application object 392
ARM EXPRESS-G (figure G.36) 2403
mapping table 846
Work_order

application assertion 416, 486, 568, 592
application object 392
ARM EXPRESS-G (figure G.17) 2384
mapping table 1309
Work_request
application assertion 399, 448, 455, 481, 537, 561, 568, 592–598
application object 394
ARM EXPRESS-G (figure G.16) 2383
mapping table 1310

year_number
AIM expanded listing types 2001
AIM EXPRESS-G types (figure H.31) 2450
ypr_enumeration
AIM expanded listing types 2001
AIM EXPRESS-G types (figure H.105) 2524
ypr_index
AIM expanded listing functions 2254
ypr_rotation
AIM expanded listing types 2001
AIM EXPRESS-G types (figure H.113) 2532