STANDARDIZED DATA EXCHANGE OF CAD MODELS WITH DESIGN INTENT

Junhwan Kim[(a)], Michael J. Pratt*[(b)], Raj G. Iyer[(c)] and Ram D. Sriram[(a)]

[(a)] National Institute of Standards & Technology (NIST), Gaithersberg, MD, USA
[(b)]LMR Systems, 21 High Street, Carlton, Bedford, MK43 7LA, UK
[(c)]TACOM, US Army, Warren, MI, USA

*Corresponding author: Tel/fax +44 (0)1234 721720, E-mail mike@lmr.clara.co.uk

ABSTRACT
Modern CAD systems generate feature-based product shape models with parameterization and constraints. Until recently, standards for CAD data exchange among different CAD systems were restricted to the exchange of pure shape information. These standards ignored the construction history, parameters, constraints, features and other elements of 'design intent' present in the model to be transferred. This paper suggests an implementational foundation for CAD data exchange with preservation of design intent, based on the use of newly published parts of the International Standard ISO 10303 (STEP). Case studies are presented which employ a hypothetical STEP application protocol (AP) using Parts 55, 108 and 111 of ISO 10303. A prototype translator based on this AP has been implemented and tested. The paper reports on the experience gained in 'intelligent' data exchange.

Key words: Data exchange, design intent, CAD models, STEP, construction history, parameterization, constraints.

1. INTRODUCTION

The exchange of CAD (Computer Aided Design) models between different CAD systems and to downstream applications has become an important industrial requirement. Until recently, national and international CAD data exchange standards, including ISO 10303 (STEP) [1,2,3], have been limited to transferring geometry. These standards have been incapable of handling the additional design intent information generated by modern CAD systems [4]. Most STEP translators can currently only transfer 'dumb' shape models representing the final result of some constructional process, all information about that process being lost in the exchange. Essential elements of the lost information include

1. *Construction history*: the procedure used to construct the shape model;
2. *Parameters*: variables associated with dimensional or other values in the model, providing an indication of what it is permissible to change;
3. *Constraints*: relationships between parameter values or between geometric or topological elements of the model, specifying invariant characteristics in the model under editing operations, usually in the interests of maintaining product functionality during modification.
4. *Features*: local shape configurations in the model that have their own semantics.

Development versions of three recently published parts of ISO 10303 were used in the work to be described:

— ISO 10303-55, 'Procedural and hybrid representation' [5] provides for the transfer of construction history information.
— ISO 10303-108, 'Parameterization and constraints for explicit geometric product models' [6] makes possible the capture and transfer of parameter and constraint information, and the representation of 2D sketches.
— ISO 10303-111, 'Elements for the procedural modelling of solid shapes' [7] provides representations for what are commonly known as 'design features'.

These documents are STEP Integrated Resources (IRs). The implementable parts of STEP are called Application Protocols (APs), and are based on specializations of selected integrated resources, as appropriate for the particular application area addressed [2,3]. In the current absence of a STEP AP referencing the three documents mentioned above, the experiments reported here used a hypothetical AP constructed from those and other IRs directly, without specialization.

It should be noted that, while this paper is concerned solely with shape models, the basic constraint definitions in Part 108 are very general, allowing specializations that can be used to model constraints of any type applying to any kind of model.

The transfer of design intent information using Parts 55, 108 and 111 of STEP allows the intuitive editing of a model in a receiving system after it has been transferred, as though it had originally been created there. The transferred model is 'intelligent', by contrast with the 'dumb' models resulting from earlier STEP exchanges [8]. In industry, much non-productive time is currently spent by CAD system users in trying to reconstruct the lost design intent in exchanged dumb models.

This paper gives theoretical foundations for model exchange with design intent between modern CAD systems, and uses them to assess the new STEP capabilities. The paper also reports on experience in the development and use of a prototype translator using the newly available ISO 10303-108 capabilities. The methodology for the exchange of parametric models using ISO 10303-108 is discussed under several sub-headings:

— **Classification:** A parametric construction history model contains many classes of information. The definition and use of a classification scheme allows us to categorize the types of information present in the model repository of the sending system. The available categories vary between systems. Classification is discussed in Section 3.

— **Structuring:** We introduce the notion of a 'unit of construction'. This will usually correspond to a design feature in the CAD system sense, but in mapping between the sending system, the ISO 10303 exchange file and the receiving system it may be necessary to include or exclude elements of supporting information for the construction of that feature, depending on how it is defined in those three places. The intention is to match the number of degrees of parametric freedom across both the pre- and post-processing phases of the exchange, and thus to break the overall transfer down into manageable units while avoiding information loss. Structuring is discussed in Section 4.

— **Interoperability:** This concerns the resolution of semantic differences between similar constructs in different CAD systems. It also covers the problem of incompatible numerical tolerances in CAD systems. The achievement of semantic and numerical interoperability will ensure the maximum preservation of model integrity and design intent in the exchange of CAD models. Interoperability is discussed in Section 5.

2. LITERATURE REVIEW

Most early work on data exchange of 3D CAD models, whether using formal (IGES, STEP) or *de facto* (DXF, SAT) standards, focused on the final geometry of the model. For example, the STEP application protocol AP203 [9] allows the transfer of boundary representation (B-rep) and closely related types of model, including assemblies of such models. The difference between AP203-based translation and the approach described in this paper is that we can now effectively transfer parameterized families of models, defined in terms of features, dimensions, constraints and construction history information — in short, the types of geometric model generated by modern CAD systems. This enables the preservation of the design intent in the original model.

An early suggestion for a method of exchanging CAD models in terms of their construction history was made by Hoffmann and Juan [10]. Their EREP (editable representation) was a specification for the representation of sequential feature-based design processes. It supported parameterization and constraints, and was the subject of a trial implementation. Another project aimed at moving beyond the exchange of pure geometric models was the PDES Inc. project ENGEN ('Enabling Next GENeration design') project [11]. This used representations based on STEP methodology [2,3]. It concentrated mainly on the exchange of geometric constraints and demonstrated the exchange of constrained 2D profile data. Bettig and Shah [12] proposed a standard set of geometric constraints for parametric modeling and data exchange. They defined explicit constraints for the relationships between all the geometric entity data types specified in ISO 10303-42, the STEP geometry/topology resource [13]. However, the STEP resource ISO 10303-108 [5,23] focuses on a smaller selection of widely implemented geometric constraints, and provides 'freeform' constraint capabilities that can be used for more specialized cases.

The work described in the present paper is aimed at preserving, as far as is possible, all aspects of design intent including relationships implied by the constructional operations used. The approach described by Rappoport [15], based on the concept of 'feature rewrites', appears to concentrate more on consistency of pure geometry between the original and the received models. One example given by Rappoport [15] concerns the replacement of an extrusion feature created by extruding up to a specified surface by an extrusion having a specified length, with an identical geometric result. This may be satisfactory from the geometric point of view, but it loses the 'design intent' characteristic that was present in the original system, where modification of the specified surface will automatically lead to a consistent modification of the extrusion.

Another approach to the exchange of procedural models, demonstrated at KAIST in Korea, uses the capture and transfer of the journal file created by CAD systems, which contains a record of every action of the system user [16,17,18]. The KAIST team has defined a non-STEP neutral format for the representation of a common command set, and in [18] has described the use of a 2-level ontology for CAD model exchange, in which the upper-level ontology plays a similar role to the STEP feature definitions used in the present work. A new STEP resource, ISO 10303-112, that allows the exchange of construction history representations of 2D profiles or sketches, has also been developed at KAIST.

ISO 10303-111 ('Elements for the procedural representation of solid shapes') [7,19], provides representations of operations for the construction of feature-based solid models. It will provide a construction history capability for Edition 2 of ISO 10303 AP203, at present the most widely used STEP application protocol. Its scope was determined by identifying the set of common capabilities of the operation sets of the most widely used commercially available CAD systems. Inevitably, some systems will have capabilities that lie outside this range. Thus complete information exchange, if possible at all, will require the representation of any such capability in terms of lower-level capabilities in the common set, with some probable loss of semantic information.

A report on the recent PDES Inc. project CHAPS ('Construction History And ParametricS') [20] provides a business case for the transfer of construction history models with and without parameterization and constraints. CHAPS concentrated mainly on using new draft ISO 10303 capabilities to transfer procedural (construction history) models defined solely by feature operations. The present paper focuses more on the transfer of procedural models containing explicitly defined parameters and constraints, and provides more technical detail of translator development. The exchange of procedural models without explicit parameters or constraints, as in the primary phase of the CHAPS project, gives some level of editability in the receiving system but omits certain aspects of design intent information that may be crucial in maintaining product functionality after a modification.

Our approach is an advance over the previous work reviewed above in several respects:

1. The exchange of construction history models based on the use of a dual model. The primary procedural model defines the construction history. It has an associated secondary model of the B-rep or some closely related explicit type. The secondary model can be used in the receiving system to check the validity of the model transfer, by comparing the reconstructed explicit model in that system, generated by evaluating the transferred primary model, with the transferred secondary model. The secondary model could also be used to resolve ambiguities, e.g., to determine which of several valid solutions of a nonlinear constraint system was chosen by the designer using the originating system.
2. The transfer of parameters and mathematical relations between parameters. In principle, any numerical attribute in the part model can be treated as a parameter in an exchange. This makes it possible, for example, to transfer a relation defining the number of hole instances in a circular hole pattern in terms of the radius of the pattern.
3. The ability to handle several types of constraints (algebraic, logical, dimensional) in a very general and easily extensible manner.
4. The preservation of multiple simultaneous parameter relationships in the model rather than a set of independent relationships. Previous work has been less general in this respect [15,18,24].
5. Conformity to the international standard ISO 10303, by using newly published parts of that standard.

3. INFORMATION CLASSIFICATION FOR STEP-BASED TRANSLATION

The classification of information concerning parameters, constraints and geometry is important because different use cases require different algorithms to handle them. Section 5 uses the criteria given in this section to show how interoperability may be achieved for different cases of parameter, constraint and geometry usage.

Some examples of STEP entities and partial STEP exchange files are given in this section and later in the paper. To understand these it is useful to know that entity types as defined in ISO 10303 are conventionally written in text in **bold_type w**ith underscores between the words forming the entity name. Exchange files contain *instances* of these entity types. Each instance has a numerical identifier, e.g., #1234, and the entity name (which appears in upper case in an exchange file) is followed by a list of values of the attributes of the entity for the particular instance concerned. It is also important to note that the work reported in this paper was carried out using development versions of the relevant ISO 10303 documents. Detail changes have been made to these documents in the meantime, but there has been no change in their basic principles.

3.1 Overview

As far as shape is concerned, Edition 1 of ISO 10303-203 (AP203 of STEP) [9] does not support the exchange of features, parameters, constraints or construction history information, but only geometrical and topological information relating to the final shape of a CAD model. For example, if a chamfer is created on an edge of a cube, the AP203 file will contain information about the chamfer face, but details of the original edge will have been lost. In the work reported here, any B-rep elements existing at intermediate stages of the design, but invisible or inactive in the final B-rep model, are accessible in the transmitted model. This is necessary because any such element may have been used as the reference element for a dimensional or other constraint, or as a reference for feature creation. The present work captures such essential aspects of the construction history of a transferred model. It also transfers details of parameterization and explicit constraints as defined in the sending system, and therefore captures additional aspects of design intent that are lost in current AP203 transfers.

In the work described here, the model exchange covers

— geometry: active or inactive, visible or invisible, selectable or un-selectable, intermediate B-rep entities and datums, of every geometric type defined in ISO 10303-42 [13];
— design features (though strictly these are nothing more than high-level definitions of local geometric configurations, having no associated application semantics, that are captured during the design process);
— construction history;
— parameterization;
— constraints: algebraic, logical and dimensional (see the definitions below).

In the absence of any currently published STEP application protocol (AP) with the required capabilities, the translators developed are based on a hypothetical AP using (among other resources) ISO 10303-55, -108 and -111 as described above, together with ISO 10303-42, the basic geometry/topology resource. For the future, Edition 2 of AP203 [19], now in preparation, will include the capabilities defined in ISO 10303-55 and -111. Those of ISO 10303-108 will probably be included in the next edition.

One important aim of this section is to clarify the meaning (for the purposes of this paper) of certain terms that are used in different and conflicting ways in various research projects and by different CAD system developers.

3.2 Parameters

3.2.1 Criteria for the classification of parameters

Terminology is often used in restricted senses in CAD system documentation, because commercial system developers' horizons are limited by the specialized functionalities of their own systems. A broad interpretation of the term *parameter* may be taken to cover a dimension of the part or one of its individual features, an entry in a part family table, or some other numerical value such as a user-defined property value. A typical narrow interpretation of the term is taken in the CAD system ProEngineer®, where it is used exclusively for a value explicitly added to the model by the user. Some systems provide a unified data structure for all parameters, while others implement different specialized data structures for specific kinds of parameters used to represent 'feature attributes or properties', 'variables' or 'dimensions', for example. Successful ISO 10303 exchange of parametric models requires all relevant parameters (in a general sense) to be extracted from a variety of data structures in the sending CAD system, with no omissions. It is necessary to distinguish different types of parameters, because different cases require different algorithms to handle them. The following list shows the criteria used for classifying parameter types in the present work. They are not mutually exclusive.

- *Criterion 1* (numeric vs. non-numeric): The dimensional characteristics of a feature can be given explicitly as numerical values, but other properties can be specified non-numerically. For example, a cylindrical hole feature may be provided with an enumerated choice of flat, hemispherical or conical bottom surface.
- *Criterion 2* (bound vs. unbound): A bound parameter is associated with (or *bound to*) an attribute of some instance in the exchange file, whose value is potentially variable following the exchange. An unbound parameter is not directly associated with an attribute in that manner, but participates in a specified mathematical relationship that may control the values of one or more bound parameters.
- *Criterion 3* (dimensional vs. non-dimensional): A dimensional parameter, which represents a dimension explicitly, is created when the designer uses a 'dimension command' as provided by the CAD system user interface. Explicit dimensions are usually associated with sketch construction, whereas feature creation operations usually give rise to implicit dimensions (see below). Some CAD systems distinguish between dimensional parameters, which they refer to simply as *dimensions*, and non-dimensional parameters, which they refer to as *parameters*. An example of a non-dimensional parameter is the number of holes in a circular pattern of holes. ISO 10303-108 treats both types of parameters in a unified way; in this context, a dimensional parameter is usually one defining a length or angle.
- *Criterion 4* (implicit vs. explicit): In either the dimensional or non-dimensional case, an explicit parameter is created when the designer uses a command for its creation, whereas an implicit parameter is automatically created as a (potentially variable) attribute of a created feature. Some CAD systems create distinct data structures for recording and handling implicit and explicit dimensions. Others do not recognize the existence of implicit parameters, regarding them as having little in common with explicitly defined parameters directly created by the user.
- *Criterion 5* (dependent, independent, free): A dependent parameter is one whose value is governed by a constraint and can only be changed by modification of independent elements in that constraint. An independent parameter is one whose value is editable and can be used to govern the values of other elements in a constraint. A free parameter is one that is associated with some attribute of the model but is not involved in any constraint.

It should be mentioned here that ISO 10303-108 defines a unified general representation for explicit parameters with real, integer, boolean or string values, whose precise application is intended to be apparent from the context in which they are used.

3.2.2 Explicit parameters

With regard to Criterion 3 above, experience in developing the proof-of-concept translators has shown the virtues of writing parameters explicitly into the neutral exchange file whenever possible. This allows the maximal preservation of an important aspect of design intent. In the ISO 10303 context, an explicit parameter is any subtype of the ISO 10303-108 **variational_parameter** entity. Most CAD systems support the explicit parameter functionality, for the following purposes:

1. ***Specification of important geometric or non-geometric properties of the part*:** This is necessary in geometric situations that are not covered by the range of feature creation operations provided. For model exchange purposes, this usage may be captured by associating instances of the ISO 10303-108 entity **variational_parameter** with dimensional or other attributes in the model and using them to control values or relationships as appropriate.
2. ***Making an implicit parameter visible to the user of the receiving system*:** This usage may be captured by simply associating an instance of **variational_parameter** with the model attribute that represents the implicit parameter;
3. ***Association of a user-defined name with an attribute or implicit parameter in the model*:** Again, an instance of **variational_parameter** may be associated with the attribute or implicit parameter, and its name attribute used to specify the desired user-defined name.
4. ***Specification of additional information for a feature attribute*:** ISO 10303-108 provides for the association of a valid domain of values for any instance of **variational_parameter**. For numerical parameters, the domain may be a continuous interval or a discrete set of values. Such domains can only be specified for explicitly defined parameters, while implicit parameters are normally restricted only as to the type of their values. Domain specification may be achieved by associating an explicit parameter with the attribute corresponding to the implicit parameter, as for the previous two cases, and setting its domain attribute as desired.
5. ***Reduction of the number of degrees of freedom of the transmitted model*:** This application is closely related to the previous one. All quantities defined in the sending system should be regarded as potentially variable, unless they are specifically

constrained to be constant. If values of dimensional or other attributes are fixed in the sending system, this fact can be captured in the exchange file through the use of the ISO 10303-108 entity **fixed_instance_attribute_set**, which would make their invariance clear to the receiving system. Effectively, this is equivalent to associating an explicit parameter with each of the attributes concerned and confining its domain to a single value.

6. ***Definition of constraint relations based on mathematical relationships*:** ISO 10303-108 provides for the representation of mathematical relationships between instances of **variational_parameter.** Some of those instances may be associated with dimensional attributes in the model. The use of ISO 10303-108 entity **free_form_assignment** or **free_form_relationship** constraints allows the specification of mathematical dependencies between the values of those attributes.

The foregoing list illustrates the flexibility of the ISO 10303-108 approach to the representation of explicit parameters.

Many CAD systems create a 'Part Family' table giving convenient access to details of independent explicit parameters in a model and any relations defined between them. If the domain of a parameter is a set of discrete values, then those values can be specified in the table. Any type of independent value may be included in a table, but dependent variables are not generally included because their values can be derived from the relations that define them. Thus the table contains a set of independent parameters defining all admissible members of the part family defined by the parametric model. A single CAD model may have several such alternative configurations associated with it, one of which will represent the 'current result', i.e., the set of parameters used to generate the current model as displayed on the screen of the sending system. Since the relevant information is available in the table, any value occurring there in the sending system can be represented as an instance of ISO 10303-108 **variational_parameter** in an exchange file. Correspondingly, parameter information transmitted in an ISO 10303 exchange file can be used by the receiving system to construct a part family table in the receiving system; it is simply a matter of organizing the information appropriately. However, it should be noted that conditional parameter relationships of the IF – THEN – ELSE type, which can be defined in the part family tables of some CAD systems, are not yet available in ISO 10303.

3.3 Constraints

Constraints specify relationships between elements of a model that are required to be maintained if the model is edited. Bettig & Shah [12] distinguish the following types of geometric constraints:

- **algebraic**, in which a mathematical relationship is required to hold between two or more parameters of the model;
- **logical**, in which a specified geometric relationship between geometric elements is required to be true;
- **dimensional**, in which a distance or angle relationship is specified between geometrical entities (possibly subject to some permitted range of variation when the model is edited).

In ISO 10303-108, most dimensional constraints are specified as subtypes of logical constraints. Thus, for example, an instance of **parallel_geometric_constraint** is a logical assertion that members of a set of lines or planes are parallel to each other, while an instance of **pgc_with_dimension** ('parallel geometric constraint with dimension') asserts that the distance between precisely two parallel elements has a specified value.

Sketch constraints, defining relationships between 2D sketch elements, may be stored by CAD systems in a specialized constraint data structure. Other types of constraints, defining inter-feature relationships or inter-part relationships in an assembly, are defined in 3D, and in the inter-feature case they usually reference datums or other auxiliary geometric elements rather than elements of the current state of the actual part geometry. There is some variation between systems in the way 3D constraints are handled. As noted below, however, the implementation described in the present paper deals only with sketch constraints.

Explicit constraints are represented as individual entities in the model, specifying relationships between two or more of its elements. Constraints of this type are created deliberately by the user, who might constrain (for example) a line to be horizontal, or two lines to be parallel, or two parameters to satisfy a specified mathematical relationship.

By contrast, an implicit constraint has no representation as an entity in the model, but occurs automatically due to the action of a constructional procedure [6,21]. During sketch creation, different CAD systems will give rise to different combinations of implicit and explicit constraints, depending, for example, on whether the system handles only fully constrained sketches or whether it allows partially constrained situations. If under-constrained situations are not allowed, the system will automatically create implicit constraints to remove any remaining degrees of freedom. Conversely, if under-constrained sketches are allowed, the designer may at some later design stage finalize the sketch by adding explicit constraints to resolve any remaining ambiguity. For the exchange of sketch information, it is fortunate that even constraints created automatically by the system are usually given explicit representations in the system database. An operation to create a fillet arc to round the junction between two line segments, for example, generally gives rise to the explicit representation of two tangency constraints referring to the arc and the two lines concerned. In some systems it is possible to turn off the automatic generation of these design intent relations. This should not be done if it may be required to transmit the model concerned to some other system, because crucial information will then be lost.

Several restrictions apply to the types of constraints handled in the implementation reported here:

1. Implicit sketch constraints have not been considered. If a sketch constraint is not explicitly represented in the CAD model in the sending system, it would be necessary to use constraint recognition to detect its presence, and experience has shown that this is a dangerous process. Intended constraints may be missed, but more importantly unintended 'constraints' may be recognized from chance geometric relationships [22]. If the CAD system allows under-constrained sketches then constraint recognition is not necessary, as mentioned above. If the system permits only fully-constrained sketches, we have assumed that every constraint, including all those created automatically by the system, will be represented explicitly in the model.

2. In the implementation described here, the only explicit constraints considered are those defined in the context of a 2D sketch. The only 3D constraints that are captured and transmitted are those implicit in feature definitions. Where datums are defined, they are transmitted in the neutral file as auxiliary geometry. Details will be given in the next section.
3. The range of constraints considered is limited to those defined in ISO 10303-108. That document includes all the most widely implemented geometric constraints, and provides a general mechanism for modelling more specialized constraints when necessary [23].
4. The work reported is confined to the exchange of single part models; assemblies have been left for the future.

3.4 Geometry

This section discusses the representation of geometry in CAD systems, and examines some associated translation issues. A classification of geometric information is proposed for distinguishing the different classes of geometric information transmitted in an ISO 10303 exchange. If a commercial CAD system does not provide adequate API functionality for accessing its internal database, alternative ways must be sought for acquiring necessary geometric information.

3.4.1 Current and invisible geometry in a CAD model

A CAD system database contains details of the geometry of all features in the model as initially created. In some systems, if a subsequent modification changes the topology of an existing feature, the geometry of the modified feature is also recorded. If a feature is suppressed (as not being necessary for some desired application) or the geometry is not relevant in the current configuration of the model, the database records the suppressed or invisible geometry. However, as it is not displayed on the CAD system screen it is not selectable by the user for modification of the model. Any created dimension, constraint or feature makes reference to one or more existing geometric elements of the model, or auxiliary elements acting as datums. This reference geometry must be visible and selectable at the time when it is referenced though it may not be visible in the final model at the time of translation, having been suppressed or overwritten by that time. The transmission of procedural or construction history models therefore requires the exchange medium to be able to reference previously selectable elements that are not selectable in the current result at the time of transfer. ISO 10303-55 supports the exchange of explicit elements for this purpose in such a way that the corresponding element of the explicit model under reconstruction in the receiving system can be identified and used appropriately as a reference element or datum in the reconstruction process.

3.4.2 Classification of geometry in construction history data exchange

The role of the pre-processor is to capture the maximum of information from the model in the sending system that will contribute towards its successful reconstruction in the receiving system. In what follows, geometric and topological entities will be denoted by G, while B_n will denote the *n*th model in the succession of 'current result' B-rep models generated by the system at the various stages of the design. In Fig. 1, the model B_1 is the result of extruding an L-shaped sketch, B_2 shows the result of adding a hole feature and B_3 the result of rounding one of the edges of B_2.

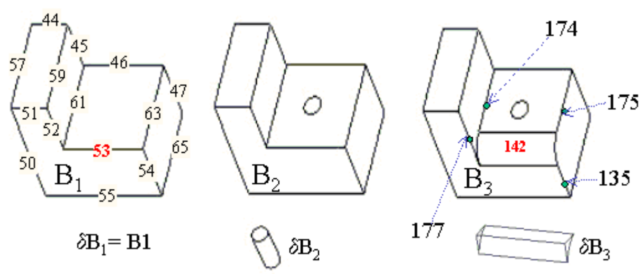

Fig.1. Example for geometry identification (L-block model)

Each feature creation operation gives rise to certain new elements associated with that feature. The only elements in the model which do not belong to features are auxiliary geometric items such as construction geometry or datums. The set of geometric or topological elements associated with the *n*th feature takes into account all differences between the *n*th secondary B-rep model and the $(n – 1)$th such model, including pre-existing elements that were modified by the latest modelling operation. If we denote the set of all B-rep geometric entities in the CAD database at stage *n* by B_n, and the status at stage $(n – 1)$ by B_{n-1} then we can define the difference as $\delta B_n = B_n – B_{n-1}$. In the example shown in Fig. 1, model B_1 has 8 visible faces and 18 visible edges (visible in the sense that they can be seen in some orientation of the model), and no invisible entities. One of the edges of B_1 is rounded at a later stage, and that edge has become invisible in model B_3.

The geometrical and topological elements associated with a single feature creation operation are divided into four classes for translation purposes:

- *Class 1*: This includes all geometric elements generated by one feature creation operation. In Fig.1, the Class 1 geometry of B_3 includes not only face 142, which was created as a result of the rounding, but all the geometric and topological elements defining delta volume δB_2 together with all the elements such as faces 135, 174, 175 and 177 which previously existed but were modified by the creation of the rounded edge. Edge 53 is needed as a reference for the creation of the rounding in the receiving system, but it is not included in the Class 1 geometry at this stage.

- *Class 2*: This class contains elements that are selected by the user from the system screen as the basis for feature operations. An example is Edge 53, which is selected at the second stage in Fig. 1 as the basis for the rounding operation whose result is evident in the third stage. All CAD systems provide the means for identification, in the system database, of specific topological elements selected in this way. In the example given, it is clear that Edge 53 is not part of the stage 3 B-rep, but it is nevertheless necessary to represent it in the procedural model to indicate unambiguously which edge was rounded.
- *Class 3*: This class consists of sketch elements. Sketches are explicitly modelled 2D shape constructs that are normally represented and managed by CAD systems separately from the 3D models in whose construction history they participate. For example, if a planar face in a B-rep model is used as the basis for a sketch and further operations then modify or delete that face, the sketch it was used to define will persist unmodified in the system database. Many dimensions and constraints applying to 3D B-rep models are originally created in 2D sketches.
- *Class 4*: This class contains elements that are not part of the feature geometry but are used as reference elements for constraining or positioning a feature, or for similar purposes in the plane of a sketch. All datums in a CAD model are stored as explicit elements that are compatible with the designer's view of the datum but may differ in detail aspects. For example, a system may use an unbounded line as a datum, but display it on the system screen as a bounded line segment.

All four classes of geometry must be transferred to enable the correct reconstruction of each feature in the receiving system. The translator implementation reported here compiles, for each feature, a list of geometry instances belonging to each class.

Examples: Edges 52, 54, 61, 63 are not elements of the Class 1 geometry for the third stage shown in Fig. 1 because they do not exist in their original forms at that stage. At the third stage, Face 142 and its edges are clearly Class 1 elements of the rounding feature, because they were created by the associated feature operation. The same applies to Edges 135, 174, 175 and 177, which are modifications of the four first stage edges listed earlier.

4. INFORMATION STRUCTURING FOR STEP-BASED TRANSLATION

In this section the 'ownership' of dimensions, constraints and parameters is defined, and a hierarchical information structure is presented. It is shown how a procedural model can be structured in terms of 'units of creation'. This enables the translation process to be expressed as a sequence of mappings of units of creation, first from the sending system to the STEP exchange file, and then from that file to the receiving system. This approach simplifies both the conception and the implementation of translators.

4.1 The ownership and structuring of identifiers

Most CAD systems maintain the sequence of operations used to create a part in a design feature tree, which is similar to a constructive solid geometry (CSG) tree. In parametric or feature-based data exchange, a given feature-based parametric history graph in a source system is mapped to a corresponding graph in the target system that generates similar (ideally, identical) geometry while preserving as much design intent information as possible [24]. For purposes of model exchange, ISO 10303-55 provides the capability for representing design feature trees.

The information written into the exchange file is derived from the model file in the originating CAD system via the system's applications programming interface (API). The model file represents a cleaned up construction history, with all extraneous operations omitted. It is inappropriate to capture the entire sequence of operations used by a designer, because not all of them are concerned with product shape. Examples of non-shape operations include the assignment of colors to elements of the shape model, or visualization commands (rotations of the model, zoom commands, etc.). Furthermore, if every detail of the design sequence were captured the resulting file would include all the errors made by the designer, and all the resulting recovery procedures, whereas what is usually needed in practice is the direct procedure leading to the final model.

The goal in the work described here is the automatic identification of 'units of construction' that have the same number of degrees of freedom in both the CAD system and the ISO 10303 neutral file. Depending on the granularity of the CAD system's representation, this may require the matching of a set of finer granularity system creation operations to a single ISO 10303 operation of coarser granularity, or vice versa. A one-to-one mapping would be ideal, but this is often impossible.

The four possibilities for identifying units of construction are shown in Fig 2, which should be regarded as applying to either of the two translation stages: (i) from the sending system to the ISO 10303 file (preprocessing) or (ii) from the ISO 10303 file to the receiving system (post-processing). The *aggregation* case combines lower-level elements into higher-level groups for transfer, while the *decomposition* case splits high-level elements into lower-level ones. Inevitably, in many cases some complex combination of these two approaches will need to be used. This emphasizes the virtues of transmitting the model in feature-based units smaller than the entire part model, because the number of possibilities for many-to-many mappings at the model level would be unsupportable for complex models made up of many thousands of elements.

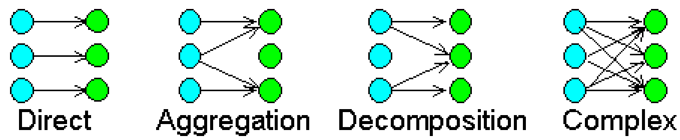

Fig.2. Mapping classes

Every transferred entity instance in an ISO 10303 exchange file is given an integer sequence number, unique to that file. CAD systems differ in the schemes they use for assigning internal identifiers to model elements. In some systems, for example ProEngineer®, each feature, topological element and geometric element has an integer identifier that is unique in the model. However, in most feature-based CAD systems, this is not the case, because the information is structured around feature entities and identifiers are only unique within a feature. Similarly, a sketch segment identifier may be unique in the model or unique only in the sketch, the latter case being more usual.

CAD system APIs provide the translator developer with access to various lists of entities occurring in the referenced model. In many cases, the lists are hierarchically structured. For example, the constraints associated with a particular feature may be found by finding the feature in the feature list and then following further pointers to the list of constraints belonging to that feature.

In the case of a sketch-based feature, sketch creation is treated as a single operation — a sketch may contain many elements, but its parametric variation is determined not by a replay of its construction history but by computation of a new solution to its constraint system.

Most CAD systems do not allow the definition of explicit constraints between elements of more than one sketch, and each sketch constraint is therefore usually unique to one sketch. A constraint between sketch elements is indirectly accessible by the model through reference to the sketch concerned, which may be considered to be a separate submodel in its own right. However, sketch elements may also be constrained with respect to datums or other geometric elements that are defined outside the sketch context and are hence more directly accessible.

To provide maximum compatibility with a wide range of CAD modelling methodologies, schemas in the ISO 10303 resource parts define entities at a very general level. In particular, there are no specialized links between the sketch and feature entities of ISO 10303-111 and the dimensional or other constraint entities defined in ISO 10303-108. The translation processors therefore need to identify the sketch or feature relationships for each constraint indirectly by comparing the geometric elements involved in that constraint with elements of sketch or feature geometry. For example, if the type and attributes of a geometric element involved in a constraint match (to within the system tolerance) with an element of a sketch occurring in the model, then we can infer that the constraint was associated with that sketch in the originating system. The most robust exchanges will be those in which all constraints can be assigned to sketches or features in this way.

The basic principle of the exchange of feature-based construction history CAD models is that each successive feature creation operation from the sending system is mapped onto a corresponding operation or sequence of operations in the receiving system. This allows us to decompose the overall process into simpler components, and to optimize the transfer of individual features. Then, ideally, following the same sequence for the mapped operations in the receiving system will result in a correct reconstruction there. It is a characteristic of this process that the partial models generated at various stages of the overall reconstruction process will also match the corresponding partial models created during the original design process.

4.2 Feature-level data exchange

Three levels of feature data exchange may be distinguished, in terms of what we refer to as their *granularity*. They are illustrated below by alternative methods for creating a block, as represented in ISO 10303.

4.2.1 Coarse granularity: (translation at the level of features as provided in the CAD system user interface).

If the feature in the sending system is exactly compatible with a construct defined in the ISO 10303 integrated resources (in particular, ISO 10303-42 for swept solids and boolean combinations, or ISO 10303-111 for more general features), the STEP construct can be used directly. The number of implicit feature parameters in the CAD system and the number of attributes of the STEP feature entity will in this case be equal. In general, constraints will be implicit, as with the parallelism and perpendicularity constraints applying to the faces of the **block** entity defined in Part 42. The exception is the use of explicit constraints on elements of a sketch to be extruded or rotated to create a feature.

In the case of a block feature, the attributes of the **block** entity in Part 42 of STEP are a position/orientation attribute of type **axis2_placement_3d** plus three dimensional attributes named **x**,**y**,**z**, of type **positive_length_measure**. If the creation of the block occurs in a sequence of operations, its dimensions will be represented by values of the dimensional attributes. Editability in the receiving system can achieved by re-running the operation sequence with changed values of those attributes. How the receiving system represents the dimensional attributes is no concern of Part 108 — the necessary information is present in the file, and can be rewritten in any way that is appropriate to that system's internal functionality.

In this simplest case, the fact that **x**,**y** and **z** are of type **positive_length_measure** implies that their values lie in the open interval $(0,\infty)$. Their values may be restricted to some smaller interval by association of explicit parameters (which have user-specified domains of validity) with the those three attributes. However, in this case the additional requirement for lower-level entities lowers the level of granularity to the mixed or intermediate case discussed below.

4.2.2 Intermediate granularity (translation using a combination of approximately compatible feature representations and lower level entities).

CAD system features frequently do not map one-to-one onto ISO 10303-42 or -111 features, because the latter represent a compromise intended to give approximate compatibility with a wide range of CAD systems. The form of the block discussed above may alternatively be created by extruding a 2D rectangular sketch with parallelism, perpendicularity and dimensional constraints applied to its sides. In such a case it will not be necessary to specify an explicit dimension for the extrusion distance, because that will be represented explicitly as the value of an attribute of the entity instance representing the extrusion. We then have a feature representation with intermediate granularity, represented by a combination of higher-level elements (the sketch, the extrusion) and lower-level elements (the constraints).

For a successful mapping between different granularities of representation, the number of degrees of freedom on both sides of the mapping must be the same. For example, the block primitive and the extruded block described above both have three parametric degrees of freedom, representing the dimensions of the block. In the first case all of them occur implicitly as dimensional parameters of the block primitive, while in the second one of them is represented implicitly as the extrusion distance and the other two as dimensions of the swept rectangle.

If the number of degrees of freedom differs, it will be necessary to use information structuring as discussed in Section 4. Each degree of freedom may be represented either implicitly or explicitly. For example, let us suppose that CAD system **A** provides an operation for creating a rectangular block primitive in a general position and orientation in model space. System **B**, by contrast, provides only an operation for creating the block in a default position and orientation, so that further operations are needed to position and orient it as desired. In such a case aggregation (as defined in Section 4) must be used in transferring the block from System **B** to System **A**. A unit of construction must be defined that combines the block primitive creation operation with the additional positioning and orientation operations, and that will have the same number of degrees of freedom as the more comprehensive (coarser granularity) operation provided by System **A**.

In the STEP context, geometric elements of supporting instances, such as the local axis systems defined by positioning and orientation operations, must be extracted from the sending system and transmitted as explicit elements in the exchange file. In the case of datum geometry used for reference purposes in such operations, different CAD systems use internal representations with different levels of detail. As much detail as possible needs to be extracted from the sending system for transmission, to allow the receiving system flexibility in the reconstruction of datums.

4.2.3 Fine granularity (translation in terms of low-level geometry and topology)

The block referred to above may in principle be exchanged as a complete B-rep model, with parallelism, perpendicularity and distance constraints applied to the planes containing its six faces, twelve edges and eight vertices. In this case, only Part 42 low-level geometrical and topological elements will be used, together with explicit constraints from Part 108. This approach is unlikely to be used for the transfer of commonly defined primitive shapes or features in a real situation, though it is useful for purposes of explanation, and may have applications in the exchange of non-standard user-defined features.

5. INFORMATION INTEROPERABILITY FOR ISO 10303-BASED TRANSLATION

In this section the interoperability of features, parameters, constraints and selected geometric elements is discussed. Our aim is the preservation of information from the sending system through the neutral file to the receiving system, despite possible major differences in the way these three systems represent that information. The following assumptions are made in our approach:

1. Only the exchange of solid models is addressed, with focus on feature-based models constructed using the capabilities of ISO 10303-111.
2. Any design intent that is not present in the model in the sending system cannot be transmitted (the alternative would require the development of general mechanisms for the automated recognition of features or other implicitly defined model characteristics from the pure construction history).
3. If the API of a particular CAD system does not support a specific necessary functionality, then we cannot exchange complete design intent. For example, at last one widely used CAD system does not allow access to constraint information in the model [20].
4. This work does not address explicit 3D constraints between part features, though that is necessary for complete exchange of design intent and will be provided in future developments.
5. ISO 10303 does not prescribe behavior of the receiving system if the transmitted model is modified there. It only assumes that it is as far as possible intuitive for the system user. The work reported here makes the same assumption.

The achievement of interoperability requires answers to questions such as the following: How can a translator determine what has been created by the designer and what has been created automatically by the system? How can it deduce explicit information needed by the receiving system if that information only exists implicitly in the sending system? How can we control the automatic generation of information in a model reconstruction following a transfer? — the presence or absence of such information influences the constructional procedure, so that apparently identical procedures do not guarantee that the same model is generated in different CAD systems.

5.1 Feature interoperability

The primary aspect of feature interoperability is that of the structuring of information, which was discussed in Section 4. However, there is also the problem that some 'features' defined in CAD systems may have no direct correspondences in ISO 10303-111. For example, in some systems the datum points, lines and planes used for dimensioning or as reference elements in constraints are regarded as features. Such elements are therefore stored in feature data structures, whereas in ISO 10303 they are classified as 'auxiliary geometric elements' as defined in ISO 10303-108. Thus an element that the sending system regards as a feature must often be written into the exchange file as a simple geometric element. The reverse may also be true in the postprocessing phase, and translators must have sufficient intelligence to make the correct mappings.

Post-processing requires the choice of the most appropriate mapping class for each feature concerned; this is determined by the granularity of the feature definition in the receiving system as compared with that of the ISO 10303-108 definition occurring in the exchange file, including all applicable dimensions and constraints.

5.2 Parameter Interoperability

Parts 55 and Part 108 together allow the transfer of hybrid model representations in which some types of information are represented explicitly and others implicitly. A parameter is an implicitly or explicitly represented variable used to control the dimensions or other gross characteristics of a model. It may be thought of as an input to a procedure, in this case a procedure that computes one instance of a family of shape models. It is worth noting that none of the three basic ways of creating a rectangular block solid, corresponding to the three levels of granularity defined above, requires the presence of explicit parameters in the model, unless it is desired to specify one or more relationships controlling its defining dimensions.

Some CAD systems give direct access to lists of parameters. However, in many cases parameters may be scattered implicitly as attributes in diverse data structures concerned with dimensions, features, mathematical relations, etc.

A CAD system may provide two or more data structures for dimensions, distinguishing between user created explicit dimensions and automatically created implicit dimensions. In such systems (for example, ProEngineer® and Solidworks®), implicit dimensions also have the effect of parameters. Implicitly defined parameters may be invisible to the designer using the graphical user interface (GUI), and only be accessible via the system API. For example, if the designer creates a fillet but does not specify a radius for it, a default value is assigned to that radius. That default value is not regarded as a dimension in the same sense as an explicitly created dimension. It is possible to change its value globally, and if that is done the radii of all fillets in the model will change. However the radius value is represented in the system, the important thing is that it exists, whether the user can see it through the GUI or not. If the system knows the value, it can be extracted and written into the exchange file.

Parameters are classified for translation purposes according to the classification of Section 3.2.1. The cases defined there are not mutually exclusive, and the classification covers all possible cases of combinations between them. More detail is given in [25]. There also arise cases of 'redundant' dimensions, which are created in the sending system purely for annotation and display purposes. Such information is irrelevant in the reconstruction of a model in the receiving system. If necessary, that system can derive dimensions as appropriate from the transferred model and reconstruct any desired annotation in its own native manner.

As regards the representation of parameters in the exchange file, the entity **variational_parameter** as defined in ISO 10303-108 can take values that are of types real, integer, boolean or string. This covers the numeric/non-numeric distinction of Section 3.2.1. ISO 10303-108 also distinguishes between bound and unbound parameters, dimensional and non-dimensional parameters, and independent and dependent parameters. Explicit parameters are represented by instances of **variational_parameter**, and implicit parameters by attributes of constructional elements, as already explained.

5.2 Constraint Interoperability

An explicit geometric constraint has three elements: (1) an identifier (ID), (2) a specification or descriptive semantics (which in dimensional cases will involve a parameter value), and (3) details of the constrained geometry. Any of these elements may give rise to constraint interoperability problems, as discussed below. In the ID case, for example, problems may arise because a constraint involving elements of a feature may have an ID that relates only to that particular feature rather than the overall model.

Commercial CAD systems represent most sketch constraints explicitly, which allows their transfer using the explicit constraint capabilities of ISO 10303-108. As already mentioned, implicit constraints arise from the constructional operations defined in ISO 10303-42, -111 and -112. They are discussed further below. Analogous construction operations in different CAD systems give rise to different results for constrained configurations in 3D; for example, in some cases implicitly created constraints are recorded explicitly in the model file, and in others they are not. Some systems store user-selected 3D reference elements for sketch constraints in 3D form; others project these reference elements onto the sketch plane if they are curves or intersect them with it if they are surfaces, and only store representations of the resulting 2D geometry.

Some CAD systems only allow binary constraints, involving just two geometric elements. Constrained configurations involving $N > 2$ geometric elements then require N - 1 binary constraints to be created. The elements involved in a geometric constraint are either *reference elements* or *constrained elements*. A reference element is a model element to which constrained elements are related by a directed constraint, while a constrained element is one that is controlled by the constraint if the model is edited following a transfer into another system. A constraint with one or more reference elements is said to be a *directed constraint*. In contrast, an *undirected constraint* has no reference elements and requires the constrained condition to hold amongst all pairs of members of a set of constrained elements [6]. ISO 10303-108 can represent *N*-ary constraints of all these types, as detailed in [23].

An *implicit* or *procedural constraint* is a constraint that results automatically from the operation of a constructional procedure used in creating a model [6, 23]. For example, a procedure 'create tangent line' may be used in a 2D procedural model. Its input is a curve (assumed to exist before the procedure is used) and a point. The output is a line through the point and tangent to the curve. In this case there is no explicit imposition of the tangency constraint by the user, and the constraint is not explicitly represented in the evaluated representation of the sketch as shown on the screen, although the *effect* of the constraint is evident in the screen display. The term *implicit dimension* has been used in a similar manner to denote a dimension that has not been explicitly created by the designer [26].

For purposes of translator development, a classification was developed for directed constraints between sketch elements. The primary distinctions relate to the reference elements, which may be (i) internal to the sketch, (ii) elements of another sketch, (iii) 3D B-rep elements external to the sketch, or (iv) datum elements external to the sketch.

5.2.1 The effect of constraint identifiers on interoperability

Commercial CAD systems associate explicit constraints with specific sketches or features, and unique identifiers are often allocated to them in the sketch or feature context. However, ISO 10303 defines no such association, and allocates unique identifiers in the exchange file at the part (or assembly) level. From the information in an ISO 10303 exchange file the feature or sketch owning the constraint must be identified as the one whose geometric elements are referenced by that constraint. Then the constraint set for a feature or sketch can be compiled by identifying all constraints referencing geometric elements associated with the construct of interest.

A sketch-based extrusion feature in ISO 10303-111 requires the underlying sketch to be provided as an instance of **face_surface.** CAD systems provide sketch-based extrusion operations, but create no constructs that map exactly to **positioned_sketch** as defined in ISO 10303-108 [6], whose specification is dictated by a requirement for upwards compatibility with other (earlier) ISO 10303 resources. The use of **face_surface** rules out the possibility that the sketch boundary is not closed, a case which requires specialized interpretation when used with an extrusion operation.

Problems will inevitably arise if inter-feature constraints are not explicitly specified. The number of pairs of elements that have to be checked for possible constraint relationships will be very large for a complex CAD model, and the possibility of spurious detection of constraints arising from chance coincidences of position or orientation of model elements will be very high [22]. Inter-feature relationships in a CAD model may be very complex if long chains of locational dimensions occur.

In the receiving system, the relation between a constraint and its owning feature must be determined through the transmitted ISO 10303 relations between the constraint and the geometric elements it involves. If the reference geometry of a constraint is a user-selected element or a datum element that is not associated with a specific sketch or feature, then it is necessary to scan all the geometry in the exchange file to find a match.

5.2.2 Semantic mapping of constraints

Semantics must be dealt with on a case-by-case basis. For example, a **vertical** direction constraint in one system may correspond to several different constraint possibilities both in another CAD system and in the ISO 10303 exchange file, as discussed below. As before, it is important to capture and transfer the maximal amount of information to allow for diverse representations of that information following the transfer.

Some possibilities for constraining a line to be vertical in a sketch include

— the use of a 'vertical' constraint, if the system provides it;
— the use of a 'same coordinate' constraint requiring the *x* coordinates of the two ends of a line segment to be equal;
— the use of a constraint requiring the line to be perpendicular to another line which is horizontal.

ISO 10303-108 does not provide a 'vertical' constraint; it would be normal to create a 2D vector or direction (0,1) in the exchange file and require the line to be parallel to that. Alternatively, a 'perpendicular' constraint could be used as in the sending system, the reference element now being a vector or direction (1,0). The effect of the 'same coordinate' constraint is also available in ISO 10303-108. In either case, a postprocessor should ideally recognize that either possibility is equivalent to a 'vertical' constraint in the receiving system, if such a constraints is provided there.

5.2.3 Constraints controlling sketch elements

Constraints may be applied between the geometric elements of a sketch, and also (in some CAD systems) between sketch elements and other geometric elements external to the sketch context. This last capability is useful, for example, in packaging applications where it is necessary to fit the parameterized shape of some component into an enclosure defined by fixed external geometry [23]. Some CAD systems allow constraints to be applied between geometric elements belonging to different sketches, but others do not.

5.2.4 Mathematical equation interoperability

Although the terminology used for algebraically defined constraints differs between CAD systems, the principles are the same. The crucial aspect of the transfer of expressions or equations is the correct mapping of the dimensional or other parameters involved. The identifiers of these parameters can be identified in the sending system by parsing the equation or expression, which is normally represented there as a character string. For reasons of upwards compatibility such constructs are represented in an ISO 10303 exchange file in terms of individual operator and operand entities, but it is easy to map between the two types of representation.

5.3 Geometric interoperability

5.3.1 The geometric accuracy problem

Differences in the internal numerical accuracy of CAD systems was a major early cause of problems in the STEP-based exchange of B-rep models [11]. For example, two points that are judged to be coincident by one system may have separate locations by the criteria of a second system. This can lead to inconsistency between the geometry and the topology of a model in an exchange from a system with loose numerical tolerances to a system with more stringent criteria. Ten years of use of AP203 for the exchange of B-reps have seen the development of techniques that greatly reduce the severity of this problem.

In a parametric model there are further possibilities for inconsistency arising from geometric accuracy problems. For example, dimensional values relating feature elements may be inconsistent with the actual geometrical definitions of those elements, or geometric constraints such as parallelism or tangency may not be satisfied exactly by the geometry as defined in the model. A CAD system will judge that a particular configuration is consistently defined if any measured inconsistency is less than the value of some geometric tolerance. However, different CAD systems use different values for their internal geometric tolerances, and some define a wider range of different tolerance types (on distances, angles, etc.) than others. The effects of geometric inaccuracies can propagate and increase in magnitude as modelling proceeds, particularly when inaccurately computed geometry conflicts with specified topology. In the transfer of boundary representation solids, topological information is usually regarded having precedence over geometry. In a similar manner, constraint information, when present, usually takes precedence over representations of individual geometric elements.

The work reported here does not address the geometric accuracy problem in any general sense, but concentrates on its effects in the procedural model as defined in the sending system. In their work on the identification and correction of errors in the exchange of CAD shape models Hong Gu et al. [27] found that structural errors and accuracy errors have a smaller effect when procedural model exchange is used than they do in the exchange of explicit B-rep models. One reason for this is that a procedural model contains no explicit Class 1 geometry. That class of geometry exists only in the redundant associated B-rep model. Thus the only explicit geometry exchanged in procedural data exchange using the hypothetical ISO 10303 Application Protocol mentioned earlier consists of sketch elements, selected elements, and auxiliary supporting or constructional elements.

A sketch constraint in the sending system may assert that one end-point of a given line coincides with one end-point of a given arc. However, the geometric elements concerned may not actually satisfy the specified constraint because of numerical errors resulting from geometric computations. The transferred model will faithfully reproduce these errors, but the presence of the constraint in the exchanged model will alert the receiving system to the fact that the two elements should coincide as specified. In principle, the constraint has precedence over the geometric information. ISO 10303-108 does not specify behaviour of the receiving system, and whether or not it rectifies the situation to restore consistency, and if so, how it performs the rectification, is a matter for that system — different systems may react in different ways. If rectification occurs, it may give rise to minor inconsistencies between the explicit models as transmitted from the sending system and as reconstructed in the receiving system.

Experience suggests the following set of guidelines for avoiding or minimizing geometric accuracy problems:

1. Use features or logical constraints wherever possible, to avoid the use of numerical values or dimensions and so reduce the possibility of accuracy problems.
2. Use the optimal setting of system-defined session parameters in creating a model that is to be exchanged with another system. For example, the user can often turn the following options on or off:
   - *the use of a grid in sketching* – this allows only the choice of a discrete set of points and angular orientations for sketch elements;
   - *the use of shaded model renderings* – this may not allow the screen selection of hidden entities in the model;
   - *the use of automatically generated constraints*, which may be accepted by the user and stored as part of the model data.

   It will be necessary to define user guidelines for each type of CAD system, recommending the most appropriate settings for the data exchange of procedural models.
3. Adopt a constructional strategy that minimizes the number of datums and user-selected reference elements in the model. The handling of such elements is one of the more difficult and computationally intensive aspects of CAD data exchange, and a simplification that does not destroy design intent will improve translation quality.

5.3.2 Preservation of selected geometry

An important problem arising in the exchange of procedural or construction history models is that of identifying, in the exchange file, model elements that were selected in the sending system by a screen pick by the system user. ISO 10303-55 provides a mechanism for the identification of selected elements that relies on the transmission of those explicit elements from the sending system in traditional ISO 10303 mode. The corresponding elements in the receiving system are then found by a geometric matching process [5,19]. Selected elements are often used as reference elements in constraints, and this application of them is discussed in Section 5.3.3. Some other considerations regarding selected elements are listed below:

- **Indirect selection of geometry in the neutral file:** This is the practice of selecting a higher-level model element indirectly in terms of one or more lower-level entities that are used to define it. Examples are the indirect selection of a face by direct selection of two of its edges, or the selection of a feature face as representative of an entire feature to be modified.
- **Automatic change of reference:** If the user references a plane as the reference element for a sketch constraint, a CAD system will usually automatically represent the constraint in terms of a compatible reference line in the sketch plane.
- **Choice of lines or planes as reference elements:** For non-sketch constraints, linear edges of the model are often used as reference elements for positioning and orienting features, rather than planes. Whether or not this is appropriate depends upon the semantics of the constraint, but in cases where no ambiguity arises, it is generally acceptable.
- **User selected elements in an exchange file:** Any element selected by the user from the CAD system screen is marked in an exchange file by instances of ISO 10303-55 entities **user_selected_elements** or **indirectly_selected_elements**, respectively, according to whether the selection is direct or indirect. Such elements will often not occur in the final B-rep, having been modified or deleted by subsequent operations.

5.3.3 Preservation of constraint geometry

For sketch constraints, explicit information concerning the geometric elements involved is usually available via the API of the CAD system. For non-sketch constraints, different systems represent the relevant information in different ways. Although some CAD systems do not allow the definition of 3D constraints within a part model, they allow the effect to be achieved indirectly, for example, by the creation of a datum element based on an element of one feature that is used in definition of a second feature. Although this implies a 3D relationship between the two features concerned, it may not be represented explicitly as a constraint in the CAD system. In an ISO 10303 exchange file, however, the possibility exists for expressing such a relationship by an equivalent explicit constraint.

Further considerations regarding the geometry of constraint elements are briefly discussed below:

- **Hidden constraint geometry in sketches and features:** It is possible for an entity that is not itself a sketch element to possess a system ID in the context of the sketch. For example, if a circular arc is defined as a visible sketch element, the underlying circle definition may be stored as an invisible element, and it is therefore possible to apply a radial dimension constraint to the circle, which will apply also to the arc. Similar situations arise in feature representations.
- **Indeterminate geometric extent of datums:** This problem often occurs when a datum element is output from the sending system to a STEP file. Datum elements are usually defined by the CAD system with default dimensions, though in some cases no dimension is specified. For example, a datum plane may be displayed by a CAD system with default dimensions of 100 × 100 units, and it may be represented internally with that precise size and the topology of a face, or as an unbounded plane that is displayed as finite for easier understanding by the designer. In either case the pre-processor should create appropriate geometry to enable the post-processor to reconstruct the relationship involving the datum correctly from the point of view of the receiving system. Initial experience suggests that the most appropriate type of geometry to transfer for a datum is the most general – for example, unbounded lines, planes, etc. rather than bounded ones.
- **Geometry in the neutral file:** ISO 10303-108 requires constraints to reference the underlying geometry of constrained topological elements (points, curves or surfaces). In some cases it is necessary to refer to the defining elements of geometrical entities rather than the entities themselves. For example, if it desired to constrain a cylindrical surface to be perpendicular to a plane it is necessary to formulate the constraint in terms of the axial direction of the cylinder rather than the actual cylindrical surface. This approach allows the total number of constraint types to be reduced because specialized constraint types do not have to be defined for each specific type of geometric entity. On the other hand, the referencing of defining elements of constrained curves and surfaces rather than the curves and surfaces themselves creates more difficulties in the implementation of translators because the references to constrained elements are indirect. An example is the ISO 10303-108 requirement to reference the underlying circle when representing a constraint on an arc lying on that circle.
- **Dimensionality of constraint elements:** ISO 10303 does not allow mixed dimensionality in any of its representations. Thus in a 3D model, a line used as a datum must have dimensionality 3, i.e. it must exist in the 3D space of the model rather than the 2D space of a sketch.
- **Representation of datums in an exchange file:** Datums will often be recorded in a procedural model as explicit points, lines or planes that provide supporting information for a constructional operation. If a datum is not part of the model geometry but is used as the reference element in a constraint used to position or orient a feature then the Part 108 entity **auxiliary_geometric_representation_item** is available for its representation.

The following guidelines have been developed for preserving design intent for constraint geometry:

1. In pre-processing, before creating reference geometry for a constraint in the exchange file, a check should be made to determine whether a geometric element that can be used for the intended purpose has already been written into the file.
2. In post-processing, avoid duplication of reference elements used in constraints; if a suitable element already exists, use it.

An example of the application of Guideline 2 above occurs when the centerline of an axisymmetric feature is used as the reference for a linear dimension, and the center-line was automatically created with the feature. No copy of that center-line should be created as a datum for dimensioning purposes; the original should be used.

In ISO 10303-42 the center-line of a feature generated by revolving a sketch is defined as an instance of **axis1_placement_2d**. This requires two elements of support geometry, of types **point** and **direction**. Often, a constraint between two such center-lines may be formulated in more than one way in the exchange file: in terms of a relation between the two defining points, or the two defining directions, or between two instances of line that can be constructed using the defining data of the two instances of **axis1_placement_2d**. The choice will depend on the CAD system generating the initial information, and it should be made in such a way that information transfer is maximized. In general, constraints formulated between elements of supporting geometry are harder to implement and interpret, while constraints between redundant constructed geometric elements result in verbose files and possible geometric inconsistency.

6. IMPLEMENTATION AND CASE STUDIES

6.1 Translator implementation

In CAD systems, most information elements in a model of a single part (dimensions, constraints, etc.) may be considered to be integral to feature definitions. The primary exceptions are (i) relations between two features, (ii) relations between elements belonging to different features, (iii) relations defined in a design family table. The procedural representation of a feature-based design is essentially a list of features, subject to the specified relations. If we use *M*, *F* and *R* to denote a model, a feature and a relation, respectively, then we can express the model file as represented in the originating system as a feature list plus a relation set:

$$M_S(F_S,R_S) = F_S + R_S = \bigcup_{i=1}^{I} F_{S\,i} + \sum_{q=1}^{Q} R_{S\,q} \quad (1.1)$$

Here the suffix *S* distinguishes information elements in the sending system, U denotes a generalized Boolean union, the result of a succession of *I* feature creation operations each performed on the pre-existing version of the part model, and Q is the total number of relations defined in the model. The feature and relation information may be further broken down as follows:

$$\begin{aligned} F_{S\,i} &= F_{S\,i}(A_i, G_{S\,i}, H_{S\,i}, D_{S\,i}, C_{S\,i}) \\ R_{S\,q} &= R_{S\,q}(A_q) \end{aligned} \quad (1.2,\ 1.3)$$

Here, for each feature, A_i denotes an attribute involved in a relation, G_i an element of geometry or topology, H_i history information, D_i a dimension, and C_i a constraint. The attributes A_q in each relation are governed by equations, usually expressed in the form of strings that need to be parsed for identification of the dimensional or other parameters they contain.

Each operation sequence is expressed in the ISO 10303 exchange file as an instance of the ISO 10303-55 entity **procedural_solid_representation_sequence**. One or more such instances can participate in an instance of the ISO 10303-55 entity **procedural_shape_representation**, which may represent an entire part history. The primary difference from the system internal representation is that the representation of explicit dimensions is external to the feature elements. Introducing new notation P_q for an explicit parameter, X_j for an explicit dimension, J for the total number of dimensions, and using the suffix *E* to denote an element of the STEP exchange model, we may express that model as follows:

$$\begin{aligned} M_E(F_E,R_E) &= F_E + D_E + R_E = \bigcup_{i=1}^{I} F_{E\,i} + \sum_{j=1}^{J} D_{E\,j} + \sum_{q=1}^{Q} R_{E\,q} \\ F_{E\,i} &= F_{E\,i}(A_i, G_{E\,i}, H_{E\,i}, C_{E\,i}) \\ D_{E\,j} &= D_{E\,j}(A_j, F_{E\,j}, X_{E\,j}) \\ R_{E\,q} &= R_{E\,q}(A_q, D_{E\,q}, P_{E\,q}) \end{aligned} \quad (2.1 – 2.4)$$

There are some general technical requirements for implementation. Temporary data structures are needed for lists of objects of various types; for example, a geometric constraint may have reference geometry of type line, arc, datum axis or datum plane, etc., so that it is convenient and efficient to implement a single data structure that can accommodate all the relevant types. This should take into account ISO 10303 supertype/subtype relationships for such elements. The pre- or post-processor then has access to all salient model elements by using pointers for traversing the list of features in the model, and the lists of dimensions, constraints and algebraic relations at the individual feature level. During translation three data structures are open simultaneously — the CAD system, the pre- or post-processor, and the neutral file. This is necessary, for example, to establish correspondences between the different identifiers used in the sending system and in the ISO 10303 exchange file for the elements involved in a constraint. Geometric elements should be translated before constraints, because they may be referenced multiple times by dimensional or other constraints, especially in the sketch context.

For some CAD systems, the use of Eqs. 2, based on the exchange file structure, may lead to a simpler translator implementation than Eqs. 1, based on the native model structure. Eqs. 2 require the generation of a temporary dimension data structure for convenience in retrieving dimensions, which is beneficial if that proves to be a frequent requirement. The problem here is related to the feature granularity issue. If there are dimensions associated with a feature in the sending system that are not implicitly accounted for in the ISO 10303-111 representation of that feature (i.e., that representation has smaller granularity), then those additional dimensions must be written into the exchange file as instances of explicit ISO 10303-108 dimensional entities. This involves a search of the model to identify the particular geometric elements related by each explicit dimension. These occur as attributes of dimensional constraint instances, and it is these attributes that must be searched. In the sending system, dimensions and other parameters, together with any equations that relate them, are often scattered between individual owning feature representations rather held centrally for the model as a whole. Thus the search process is greatly simplified by the generation of a temporary array of dimensions for the model as a whole, providing capabilities for indexing dimensionally constrained geometric elements to individual features.

For two-way exchange between two CAD systems A and B four processors are needed: a preprocessor and a post-processor for both systems. All the processors developed have the same structure, paralleling the basic structure of a typical CAD system API. The same structure can therefore be used for developing processors for other CAD systems, or can form the basis of a standardized high level API for developing ISO 10303 translators. In the work describe a parser was specially developed to handle STEP file parsing, but a commercial parsing library could alternatively be used.

6.2 Case Study

Experiments in model transfer were performed with several test parts, all taken from ProEngineer® tutorial material and related sources. All the transferred models were found to exhibit satisfactory variation under editing in the receiving system. Space permits only one case study to be described here; others are detailed in [25].

It is first necessary to explain the structure of the ISO 10303 exchange file as defined in ISO 10303-21 (‘Clear text encoding of the exchange structure’). A STEP Application Protocol, relating to a specific product type and product life-cycle stage, defines the entity types that occur in an exchange. Each defined entity is characterized by a set of attributes, which may have values of various types. The exchange file consists primarily of a set of instances of the defined entity types, each instance having a unique identifier in the file which has the form of a integer number preceded by a hash character ‘#’. Each instance is specified by the name of the entity type concerned (in upper case) followed, in parentheses, by the list of values of the attributes for the particular instance concerned. If these attributes have ‘simple’ types (real, integer, Boolean or character string, for example) they are given explicitly. However, some attributes may have values defined by other instances in the file, and in that case they are specified in terms of the relevant instance identifier. For example, an unbounded line is defined in terms of a point and a direction. A line instance in the file will therefore contain references to an instance of a point and an instance of a direction. It is hoped that this explanation will make the following examples reasonably self-explanatory. Many details have been omitted, and ISO 10303-21 should be referred to for further information. Although ASCII file exchange is used for illustrative purposes in the case studies, it should be noted that ISO 10303 provides several other methods for the exchange and sharing of product information.

The part for the case study has explicit geometric constraints and dimensions. Its base shape is a linear extrusion of a sketch, and two features are defined upon it, a circular protrusion and a circular depression, both having the same radius. In the originating system (ProEngineer®), the constraints defined are 5 point coincidences, 3 tangencies, 2 horizontal direction constraints and 5 dimensions. Additionally, there is one ‘same (*x*) coordinate’ constraint between the center point of the arc R92 and the lower end point of that arc.

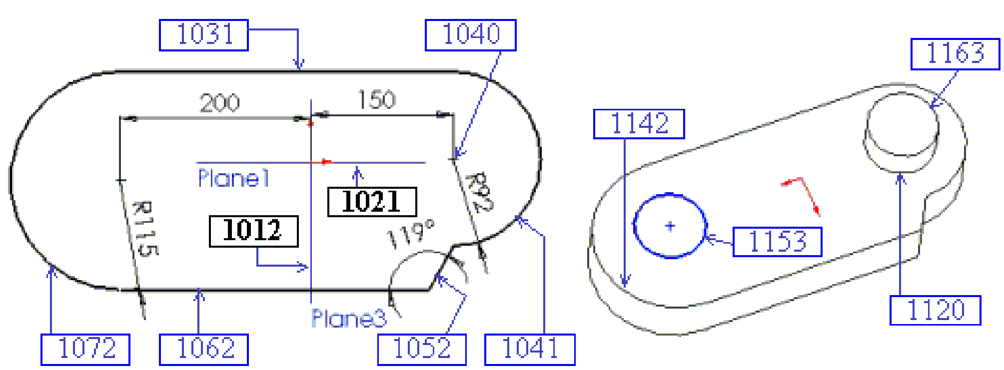

Fig 3. Sketch segment example

Table 1. Fragment of ISO 10303-21 exchange file for the constraints in the sketch shown in Figure 3

1-1	#1081= TANGENT_GEOMETRIC_CONSTRAINT (‘’,(#1031,#1041),( )); #1082= TANGENT_GEOMETRIC_CONSTRAINT (‘’,(#1031,#1072), ( )); #1083= TANGENT_GEOMETRIC_CONSTRAINT (‘’,(#1062,#1072), ( ));
1-2	#1084= PARALLEL_GEOMETRIC_CONSTRAINT (‘’,(#1031),(#1021)); #1085= PARALLEL_GEOMETRIC_CONSTRAINT (‘’, (#1062),(#1021));
1-3	#1088= RGC_WITH_DIMENSION (‘’,(#1041), 92.0); #1089=AGC_WITH_DIMENSION(‘’, (#1052, #1062), 119); #1090= RGC_WITH_DIMENSION (‘’,(#1072), 115.0);
1-4	#1092= PDGC_WITH_DIMENSION(‘’, (#1071), (#1012),200.0);
1-5	#1091= PDGC_WITH_DIMENSION(‘’, (#1040, #1037), (#1012),150.0);
1-6	#1125=RGC_WITH_DIMENSION(‘’,#1120, 50.0);
1-7	#1167=COAXIAL_GEOMETRIC_CONSTRAINT(‘’,(#1142),(#1153),()); #1168=RADIUS_GEOMETRIC_CONSTRAINT (‘’,(#1163,#1153),());

A fragment of the ISO 10303 exchange file for the object shown in Fig. 3 is given in Table 1. This lists some of the instances referred to below; others are indicated in the figure. Five incidence constraints (not shown) apply to endpoints of successive curve segments making up the initial sketch, to ensure that it defines a continuous closed boundary. The three tangent constraints in the table ensure smooth junctions between line segments and arcs in the sketch. ISO 10303-108 does not define the ‘horizontal’ constraint available in the CAD system, and the effect is achieved here by constraining the two long line segments to be parallel to #1021, a line instance (shown in Fig. 3) that has the required direction, using **parallel_geometric_constraint**. Section 1-3 of Table 1 contains the dimensional constraints for the arc radii and the angle specified in the sketch. Section 1-4 contains the dimensional constraint specifying the distance from the center point #1071 of the arc #1072 to the datum plane 3 (#1012).

In all the above cases ISO 10303-108 provides constraints that are compatible with the constraints in the model in the originating system. However, section 1-5 of Table 1 shows an example of an aggregation mapping, necessary because of differences in granularity. An instance of the constraint **pdgc_with_dimension** (point distance constraint with dimension) is shown, with reference element #1012, constrained elements #1037, #1040, and distance value 150. This constraint requires that the center and lower end-points of the arc are both distant 150 units from the vertical line #1012. This is compatible with, but not identical with, the intention in the originating system, which imposed a dimensional and a same-coordinate constraint. Thus, in this case two constraints in the sending system have been mapped to a single constraint in the ISO 10303 exchange file.

In Section 1-6 of Table 1, instance #1125 represents an explicit radial dimension for the profile #1120 of the circular protrusion. In Section 1-7, instance #1167 constrains the hole cross-section #1153 to be coaxial with the cylindrical end #1142 of the base solid, represented by arc #1142. Finally, instance #1168 constrains the radii of the hole and the boss to be equal. The constraint instance #1168 requires the radii of the circle instances #1153 and #1163 to be equal.

6.2 Part family transfer

We conclude this section with an illustration of a master model (see Fig. 4) as transmitted between systems, and some members of the same part family generated in the receiving system by editing values of some of the ten independent parameters in the model. Some dependent parameters in the model were correctly reevaluated in the receiving system according to transmitted constraint relationships. All degrees of freedom in the model were tested, and no incorrect results were generated. This case study demonstrates the outstanding advantage of data exchange using the new STEP capabilities — the ability to modify a model after transmission, in accordance with the designer’s original intentions.

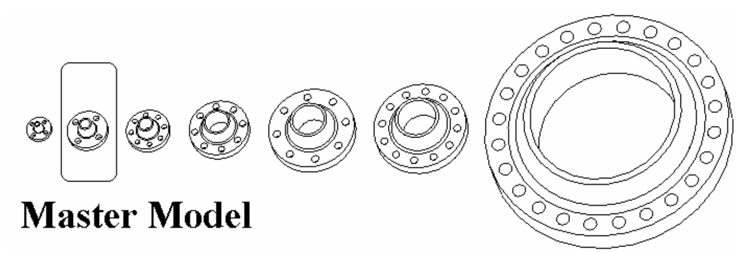

Fig 4. Results of demonstration of part family transfer

7. CONCLUSIONS

This paper presents foundations for the standardized inter-system exchange of construction history shape models with parameterization and constraints. These foundations have been used in a proof-of-concept test of the exchange of such models, using recent enhancements to ISO 10303, the STEP standard. The coverage of geometry and design intent achieved is admittedly only a subset of what will be required for full practical translators, and not all problems encountered have been fully overcome. However, we must recall that after the initial publication of Edition 1 of STEP in 1994 several years of experience were required before exchanges of boundary representation models became reliable. The exchange of procedural or construction history models is a new departure for STEP, and a comparable learning period must be expected. This will start late in 2007 with the publication of Edition 2 of AP203, containing construction history capabilities.

The APIs of commercial CAD systems are not primarily intended as an interface for the model exchange, and it may not always be possible to transfer all the design intent information associated with a model. The present paper makes recommendations for achieving the maximal possible information transfer, and it is believed that, subject to the general acceptance of some limitations, useful practical translators can be implemented that will realize the business advantages spelled out in [20].

Future research will include the data exchange of feature-based assemblies between different CAD systems. This will involve models containing explicit constraints in 3D space. It is also intended to adopt an ontological approach for the semantic mapping of modeling elements between CAD systems, based on the methodology outlined by Patil et al. [28].

DISCLAIMER

Certain commercial software systems are identified in this paper. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology (NIST); nor does it imply that the products identified are necessarily the best available for the purpose. Further, any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NIST or any other supporting US government or corporate organizations.

REFERENCES

[1] ISO 10303. Industrial automation systems and integration – Product data representation and exchange – ISO 10303:1994. Geneva, Switzerland: International Organization for Standardization (ISO), 1994. Note: the ISO catalogue is online at http://www.iso.ch/cate/cat.html – search on 10303 for a listing of parts of the standard.

[2] Owen J. STEP: An Introduction. 2nd ed. Winchester, UK: Information Geometers; 1997.

[3] Pratt MJ. Introduction to ISO 10303, the STEP standard for product data exchange. J Computing and Information Science in Engineering, 2001;1:102-103.

[4] MacKrell J. Exchanging product design data: business benefits of the collaboration gateway. CIMdata, http://www.CIMdata.com, April 2004

[5] ISO 10303-55. Industrial automation systems and integration – Product data representation and exchange: Integrated generic resource: Procedural and hybrid representation. Geneva, Switzerland: International Organization for Standardization, 2005.

[6] ISO 10303-108. Industrial automation systems and integration – Product data representation and exchange: Integrated application resource: Parameterization and constraints for explicit geometric product models. Geneva, Switzerland: International Organization for Standardization, 2005.

[7] ISO 10303-111. Industrial automation systems and integration – Product data representation and exchange: Integrated application resource: Elements for the procedural modelling of solid shapes. Geneva, Switzerland: International Organization for Standardization, 2007.

[8] Wang Y. Constraint-enabled design information representation for mechanical products over the internet. Ph.D thesis, University of Pittsburgh, 2003.

[9] ISO 10303-203. Industrial automation systems and integration – Product data representation and exchange: Application protocol: Configuration controlled design of mechanical parts and assemblies. Geneva, Switzerland: International Organization for Standardization, 1994.

[10] Hoffmann CM, Juan R. EREP: An editable high-level representation for geometric design and analysis. In: Wilson PR, Wozny MJ, Pratt MJ, editors. Geometric Modeling for Product Realization, Amsterdam:North-Holland; 1993, p. 129-164.

[11] Shih C-H, Anderson WD. A design/constraint model to capture design intent. In: Proc. 1997 ACM Solid Modeling Symposium, Atlanta, GA, USA, NewYork: ACM Press, June 1997.

[12] Bettig B, Shah JJ. Derivation of a standard set of geometric constraints for parametric modeling and data exchange. Computer-Aided Design 2001;33:17-33.

[13] ISO 10303-42. Industrial automation systems and integration – Product data representation and exchange: Integrated generic resource: Geometric and topological representation. Edition 3. Geneva, Switzerland: International Organization for Standardization, 2003.

[14] Pratt MJ. Extension of the STEP standard for parametric CAD models. J Computing and Information Science in Engineering 2001;1:269-275.

[15] Rappoport A. An architecture for universal CAD data exchange. In: Proc. 2003 ACM Solid Modeling Symposium, Seattle, WA, USA, New York: ACM Press, June 2003, p. 266-269.

[16] Choi G-H, Mun D, Han S. Exchange of CAD part models based on the macro-parametric approach. Int J CAD/CAM 2002;2: 13-21.

[17] Mun D, Han S, Kim J, Oh Y. A set of standard modeling commands for the history-based parametric approach. Computer Aided Design 2003;35:1171–1179.

[18] Seo T-S, Lee Y, Cheon S-U, Han S, Patil L, Dutta D. Sharing CAD models based on feature ontology of commands history. Int J CAD/CAM 2005;5.
[19] Pratt MJ, Anderson WD, Ranger T. Towards the standardized exchange of parameterized feature-based CAD models. Computer Aided Design 2005;37:1251-1265.
[20] Stiteler, M. Construction History And ParametricS: improving affordability through intelligent CAD data exchange. CHAPS Program Final Report, Advanced Technology Institute, 5300 International Boulevard, North Charleston, SC 29418, USA, January 2004.
[21] Pérez A, Serrano D. Constraint based analysis tools for design. In: Proc. 1993 ACM Solid Modeling Symposium, Montreal, Canada, New York: ACM Press, 1993, p. 281-290.
[22] Anderl R, Mendgen R. Parametric design and its impact on solid modeling. In Proc. 1995 ACM Solid Modeling Symposium, Salt Lake City, UT, USA, New York: ACM Press, 1995.
[23] Pratt MJ. A new ISO 10303 (STEP) resource for modeling parameterization and constraints. J. Computing and Information Science in Engineering 2004; 4;339-351.
[24] Spitz S, Rappoport A. Integrated feature-based and geometric CAD data exchange. In: Proc. 2004 ACM Solid Modeling and Applications Symposium, New York: ACM Press, 2004.
[25] Kim J, Pratt MJ, Iyer RG, Sriram RD. Data Exchange of Parametric CAD Models using ISO 10303-108. NIST Intergovernmental Report, in preparation. Gaithersburg, MD, USA: National Institute of Standards and Technology.
[26] Shah JJ, Mäntylä M. Parametric and feature based CAD/CAM: Concepts, techniques and applications. New York: John Wiley & Sons; 1995.
[27] Gu H, Chase TR, Cheney DC, Bailey TT, Johnson D. Identifying, correcting, and avoiding errors in computer aided design models which affect interoperability. J. Computing and Information Science in Engineering 2001;1:156-166.
[28] Lalit Patil, Debasish Dutta and Ram Sriram, *Ontology-based exchange of product data semantics*, IEEE J. Automation Science & Engineering **2**, 213-225, 2005.

Author biographies:

Junhwan Kim has spent three years as a Guest Researcher in the Design & Process Group of the Manufacturing Systems Integration Division at the US National Institute of Standards and Technology (NIST) in Gaithersburg, MD. Kim received his BS (1995), MS(1998), and PhD (2003) in Mechanical Engineering from KAIST, the Korea Advanced Institute of Science and Technology. His research interests include CAD databases, CAD data exchange, knowledge-based design, virtual reality and ubiquitous computing.

Mike Pratt is a Principal Consultant with the partnership LMR Systems. Previously he worked for Rensselaer Polytechnic Institute in Troy, NY, USA, spending much of his time as a Guest Researcher at the US National Institute of Standards and Technology (NIST) in Gaithersburg, MD. Earlier he was Professor of Computer Aided Engineering and head of mathematics and computer science at Cranfield University in the UK. His research interests include all aspects of product modelling in mechanical engineering, and especially its uses in the integration of CAD with downstream applications. He is active in the development of the STEP standard, as leader of the ISO TC184/SC4 Parametrics Group. Pratt has an MA in physics from Oxford University, an MSc in aeronautical science and a PhD in mechanical engineering from Cranfield University. His professional memberships include BCS, IMA and RAeS in the UK, and IFIP Working Group 5.2 (Computer Aided Design). He is on the editorial boards of the journals Computer Aided Geometric Design, International Journal of Shape Modelling and International Journal of Product Lifecycle Management.

Raj G. Iyer leads the Product Lifecycle Management program at the US Army's Tank Automotive Research development and Engineering Center (TARDEC) where is responsible for developing and integrating technology and processes for the lifecycle data management of the Army's combat and tactical vehicle fleet. He has over 15 years of academic, private sector and federal Government experience relating to product data interoperability and standards relating to CAD/CAM, PDM and PLM. He has authored over two dozen publications on the subject and has presented papers at numerous conferences worldwide. Iyer supports several federal agencies such as the Office of Secretary of Defense and NASA as a PLM subject matter expert. He is the DOD's representative to the ISO TC184/SC4 Industrial Data standards committee as well as the Army's representative to the Executive Board of Directors at the PDES Consortium. He also serves on the editorial committee of the International Journal of PLM. Iyer has an M.S. and Ph.D. in Electrical Engineering from the University of Texas, Arlington, USA and a B.E. from REC, Tiruchy, India.

Ram D. Sriram leads the Design and Process group in the Manufacturing Systems Integration Division at the US National Institute of Standards and Technology, where he conducts research on standards for interoperability of computer-aided design systems and on healthcare informatics. Prior to that he was on the engineering faculty (1986-1994) at the Massachusetts Institute of Technology (MIT) and was instrumental in setting up the Intelligent Engineering Systems Laboratory. At MIT, Sriram initiated the MIT-DICE project, a pioneering project in collaborative engineering. He has co-authored or authored over 200 publications in computer-aided engineering, including several books, and was a founding co-editor of the International Journal for AI in Engineering. In 1989, he was awarded a Presidential Young Investigators Award from the National Science Foundation, U.S.A. Sriram is a Fellow of the American Society of Mechanical Engineers and a Senior Member of the Institute of Electrical and Electronics Engineers. He has a B.Tech. from IIT, Madras, India, and an M.S. and a Ph.D. from Carnegie Mellon University, Pittsburgh, USA.