CAD Reusability and the Role of Modeling Information in the MBE Context

Jorge D. Camba
jdorribo@uh.edu

Manuel Contero
mcontero@upv.es

Pedro Company
pcompany@uji.es
Agenda

01 Context and Background
- MBE
- Digital Thread

02 Key Terms
- Product Model
- Communication
- Reuse

03 Software Analogy
- Source Code
- Documentation
- Software Reuse

04 Vision
- Quality Modeling
- Quality Testing
- Model Enrichment

05 Contributions
- Demo
Model-Based Enterprise

- A Model-based Enterprise (MBE) is an organization that applies modeling and simulation technologies to integrate and manage all of its technical and business processes related to production, support, and product retirement\(^1\).

- It applies product and process models at every step of the product life-cycle.

- The core MBE tenet is that **data is created once and directly reused** by all data consumers\(^1\).

---

The Digital Thread for Smart Manufacturing project enables the repurposing, reuse, and traceability of information throughout the product lifecycle. Research focuses on standards and implementation needs to exchange information between each phase of the lifecycle, particularly between engineering, manufacturing, and quality functions.

1NIST, Systems Integration Division. *Enabling the Digital Thread for Smart Manufacturing* (https://go.usa.gov/6nPh)
“Over-the-wall” Design

Key Terms

Product Model

Communication

Reuse
Key Terms

An Analogy: Software Engineering

Product Model

Communication

Reuse

Source Code

Documentation

Software Reuse
An Analogy: Software Engineering

Software Engineering Sub-Disciplines

- Software requirements
- Software design
- Software construction
- Software testing
- Software maintenance
- Software configuration management
- Software engineering management
- Software development process

- Software engineering models and methods
- Software quality
- Software engineering professional
- Software engineering economics
- Computing foundations
- Mathematical foundations
- Engineering foundations
Software Quality

- The way a program is written can have important consequences.

- Software quality refers to:
  - Functional quality
  - Structural quality

- International standard for the evaluation of software quality: ISO/IEC 25010:2011
  - Functionality, Reliability, Usability, Efficiency, Maintainability, and Portability

- Examples of Strategies:
  - Coding conventions
  - Code refactoring
  - Programming Paradigms (Object-Oriented, Structured, etc.)
• Just like the quality of a software system depends on the correctness and efficiency of its code, the quality of products depends on the quality of their design processes, which then depend on the quality of their data.

• **Product Data Quality (PDQ)** is a measurement of the accuracy and appropriateness of all data involved in the design and manufacturing of a product combined with the timeliness with which they are provided to the stakeholders who may need them\(^1\),\(^2\).

• Current Model Quality Tools (MQT) are primarily aimed at preventing “easily” solvable low semantic level mistakes and incoherencies\(^3\).


Product Data Quality Model¹

- **Morphological quality**: related to the geometrical and topological correctness of the CAD model

- **Syntactic quality**: evaluates the use of proper modeling conventions.

- **Semantic/pragmatic quality**: takes into account the model's capability for reuse and modification

---

An Analogy: Software Engineering

Communication

- In software engineering, developers use source code comments to support program maintenance and future modifications.

```
Source code example from Wikipedia
/* loop backwards through all elements returned by the server (they should be processed chronologically)*/

for (i = (numElementsReturned - 1); i >= 0; i--)
{
    /* process each element's data */
    updatePattern(i, returnedElements[i]);
}
```

- Tools such as Doxygen, Javadoc, Apple’s HeaderDoc, or Microsoft’s Sandcastle can automatically generate project documentation from source code comments.
• **Product and Manufacturing Information (PMI)**
  
  – Geometric Dimensioning and Tolerancing (GD&T)
  – 3D annotations (text)
  – Surface finish
  – Material specifications

• **Standards for 3D part definition:**
  
  – ASME Y14.41
  – ISO 16792
Software Reusability

- “Code reuse is the Holy Grail of Software Engineering.”
  - Douglas Crockford (Senior JavaScript Architect at PayPal)

- Reusability is the use of existing assets in some form within the software product development process, including code, software knowledge, test suites, designs and documentation.

- Examples of Strategies:
  - Subroutines
  - Software Libraries and Components
  - Design Patterns
Design Reuse

- Standard Parts and Libraries
• All engineering organizations report that they reuse designs, but the top performing ones deploy techniques and technologies to CAPITALIZE on design reuse.

• Design Reuse \rightarrow Model Reuse


Design Reuse

- There are important obstacles to implement CAD model reuse

<table>
<thead>
<tr>
<th>Challenges</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model modification required expert CAD knowledge</td>
<td>57%</td>
</tr>
<tr>
<td>Models are inflexible and fail after changes</td>
<td>48%</td>
</tr>
<tr>
<td>Only original designer can change models successfully</td>
<td>40%</td>
</tr>
</tbody>
</table>

Design Reuse

- There are important obstacles to implement CAD model reuse

<table>
<thead>
<tr>
<th>Challenges</th>
<th>Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model modification required expert CAD knowledge</td>
<td>Train users to increase CAD skills</td>
</tr>
<tr>
<td>Models are inflexible and fail after changes</td>
<td>Design for wide range of modifications</td>
</tr>
<tr>
<td>Only original designer can change models successfully</td>
<td>Detail design information in model (PMI)</td>
</tr>
<tr>
<td>57%</td>
<td>71%</td>
</tr>
<tr>
<td>48%</td>
<td>64%</td>
</tr>
<tr>
<td>40%</td>
<td>36%</td>
</tr>
</tbody>
</table>

Vision

- Improving digital *product representation* and *data quality* and its connection to *design knowledge* at the early stages of the product life-cycle can reduce the complexity of the “digital thread” and lead to more efficient product development processes.

- The execution of this vision is based on three primary ideas:

  - High Quality Modeling
  - Model Quality Testing
  - Model Enrichment
1. High Quality Modeling

- History-based parametric CAD modeling
1. High Quality Modeling

• Rebuild Errors
1. High Quality Modeling

Source: Evan Yares, *The failed promise of parametric CAD* (http://www.3dcadworld.com/the-failed-promise-of-parametric-cad/)

© Roger Penwill
1. High Quality Modeling

Design Intent

- In parametric CAD, design intent is expressed implicitly in the model's design tree.

- The size and underlying complexity of the design tree can grow rapidly, even for relatively simple models.

- Modeling decisions are not explained.
1. High Quality Modeling

- History-based parametric CAD modeling (behind the scenes)

Total Features: 48
Total Dependencies: 97
1. High Quality Modeling

- Parent/child relationships can be problematic.

- The interdependencies of the model should be defined according to the design intent.

- But the design tree is only a sequential list of features…
1. High Quality Modeling

Strategies

- Improving the understanding of parent-child relationships

Solidworks
1. High Quality Modeling

Strategies

- Improving the understanding of parent-child relationships

PTC Creo
1. High Quality Modeling

Strategies

• Improving the understanding of parent-child relationships

Siemens NX
1. High Quality Modeling

Strategies

- Improving the understanding of parent-child relationships
1. High Quality Modeling

Strategies

- Improving the understanding of parent-child relationships

Design Tree

<table>
<thead>
<tr>
<th>Flange</th>
</tr>
</thead>
<tbody>
<tr>
<td>REVOLVE</td>
</tr>
<tr>
<td>Sketch 1</td>
</tr>
<tr>
<td>CUT</td>
</tr>
<tr>
<td>Sketch 2</td>
</tr>
<tr>
<td>CIR. PATTERN</td>
</tr>
<tr>
<td>FILLET</td>
</tr>
</tbody>
</table>

Dependency Matrix

<table>
<thead>
<tr>
<th>Child</th>
<th>Revolve</th>
<th>Cut</th>
<th>Cir. Pattern</th>
<th>Fillet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revolve</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Cut</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Cir. Pattern</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fillet</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
1. High Quality Modeling

Strategies

• Modeling methodologies
1. High Quality Modeling

Strategies

• Modeling methodologies

“I'M A STRONG BELIEVER IN CAD STANDARDS - THAT'S WHY I USE THIS EXCELLENT SET OF MY OWN”

“WE BELIEVE WE NOW HAVE A CONSENSUS ON THE CAD STANDARDS”

© Roger Penwill
1. High Quality Modeling

Strategies

• Modeling methodologies

Horizontally-Structured CAD/CAM Modeling


Explicit Reference Modeling


Resilient Modeling

1. High Quality Modeling

- Design trees by methodology
1. High Quality Modeling

- Graph complexity and structure varies by methodology

Horizontal

Explicit References

Resilient

reference geometry  model features
Contributions

• We have conducted numerous experiments to test and compare different modeling approaches in a variety of design scenarios and modeling situations.

• Levels of quality (Linguistic Model):
  – Morphological, syntactical, and semantic.

• Dimensions of quality:
  – Validity, completeness, consistency, conciseness, clarity, and design intent.

• The goal is to develop a defined set of quantitative metrics.
2. Model Quality Testing
Analyze, correct, and repair defects (particularly at the semantic level) is still necessary to guarantee the reliability and robustness of master digital models.
2. Model Quality Testing

Model Quality Testing tools

- Quality testing tools for model verification, validation, and comparison are essential, as exporting CAD models that contain errors or anomalies to different downstream applications is prone to data corruption, which typically requires the models to be reworked by the downstream user.

SolidWorks Design Checker

NX Checkmate
Model Quality Testing Contributions

- DSM Model analyzer for Solidworks.
2. Model Quality Testing

Model Quality Testing Contributions

- Management of inherited dependencies in DAG.
Model Quality Testing Contributions

• Detection of Dimensions of Quality.
• Automatic Repair and Correction.
3. Model Enrichment

- Enrich models with structured design information and mechanisms to centralize knowledge and facilitate design communication.

- Annotated models provide significant benefit over non-annotated models in terms of reusability, especially when undergoing design alteration.

- Built-in historical record of design information.
3. Model Enrichment

Challenges

- *Representation*: external vs. internal.
- *Content and Structure*: what needs to be provided and in what form.
- *Interface*: interaction system used to enter and retrieve annotation information.
- *Visualization*: visual clutter, information overload, etc.
- *User Motivation*: incentives, documenting, etc.
3. Model Enrichment

Representation

- **Internal**: annotations stored within the CAD model
- **External**: annotations stored in external repositories

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal</td>
<td>• Easy implementation</td>
</tr>
<tr>
<td></td>
<td>• Full integration with the model (low maintenance)</td>
</tr>
<tr>
<td></td>
<td>• Efficiency in terms of processing and manipulation</td>
</tr>
<tr>
<td></td>
<td>• Already supported by most CAD systems</td>
</tr>
<tr>
<td>External</td>
<td>• Original document changed</td>
</tr>
<tr>
<td></td>
<td>• Difficulty for multiple sets of markups</td>
</tr>
<tr>
<td></td>
<td>• Difficult to search annotations content</td>
</tr>
<tr>
<td>Multiple independent sets of markup</td>
<td>• Difficulty of implementation</td>
</tr>
<tr>
<td>Progressively information update</td>
<td>• Problem of persistent references</td>
</tr>
<tr>
<td>Easy distribution of information</td>
<td>• File maintenance</td>
</tr>
<tr>
<td>in collaborative environments and</td>
<td></td>
</tr>
<tr>
<td>over the web.</td>
<td></td>
</tr>
<tr>
<td>Information can be processed and</td>
<td></td>
</tr>
<tr>
<td>analyzed separately</td>
<td></td>
</tr>
</tbody>
</table>
3. Model Enrichment

Content and Structure

• What information should be included? How do we capture it?
• Structured vs. unstructured
• When is the user providing too much information?
• Can we develop a standard structure or language for annotations?
3. Model Enrichment

Interface

• Mechanisms to support user interaction with the information are needed.

• None of the current standards provide explicit guidelines about managing 3D annotations, although they do recommend the use of groups, layers, or views to make the model more readable.

• Interaction with annotations should be integrated with the CAD application (PMI modules) so designers do not need to leave the CAD environment to annotate the models.

• Effort (time) vs. Benefits
3. Model Enrichment

Interface in current systems

- Current systems: poor usability. Basic information grouping tools.
- No selective visualization.
- No filtering and/or search functionality.
- Disconnected from collaborative design tools.
- No traceability of information and how it evolves over time.
- No links with external documents and information systems.
3. Model Enrichment

Visualization

Refer to Model AAC314 for weight control information.

Dimensional adjustments for weight control must be made in unmachined areas.

Thickness increased after initial FEA test results.

Ribs on both sides may require update when side cuts are modified.

Ensure a minimum angle of 20 degrees.

Angle reworked to 20 degrees.

Stamp id number on this sheet following standard formatting.

Modify extrude offset if overall depth changes to ensure ribs remain centered.

Hole distribution defined individually by size.

Do not pattern or use symmetry.

Ensure standard metric sizes for all holes.

Optional element according to client specification.

Max angle (49 deg). Do not increase.

Break all sharp edges to R3.

Ensure correct dimensions of rib and fillets if modifying this cut.

Edit offset in Extrude to modify spacing.

Ribs from two extrusions with same sketch.

Ribs sketched on construction plane. Perpendicularity depends on proper angle of plane.

Edit geometric constraint in sketch to modify overall angle.

Modify fillet radius if body angle changes.

Ensure cut doesn’t interfere with fillet.

Do not use symmetry. Angles may vary.

Optional for weight control.

Contact design center if modifying M4 holes.

3.00 x 3.00 Minimum cross section defined by FEA analysis.
3. Model Enrichment

User Motivation

• Designers are reluctant to spend time annotating their designs

• The designer has no further use of annotations, as she already understands the design. Why should the designer do something that is only beneficial for people that come after her?
3. Model Enrichment

Contributions: Annotation Manager

- CAD Integration
- Enhanced User Interface
- Advanced visualization, filtering and search functionalities
- Interactive visualization via model features
3. Model Enrichment

Contributions: Annotation Manager

- External document support (hyperlinks, multimedia content, etc.)
- PLM Integration
- Historical records and information evolution
- Videoconferencing
3. Model Enrichment

Demo
Thank You

Jorge D. Camba
jdorribo@uh.edu

Manuel Contero
mcontero@upv.es

Pedro Company
pcompany@uji.es

The icons used in this presentation are courtesy of José Sanmartín González, Delwar Hossain, Justin Blake, Hea Poh Lin, and Creative Stall from Noun Project (https://thenounproject.com)