Understanding Quality in Conceptual Modeling

Odd Ivar Lindland, Guttorm Sindre, Arne Sølvberg
IEEE Software, March 1994
Contents

What are the properties of a good model?

A framework for quality considerations

The quality framework

Quality types and goals

Techniques and tools for improved quality

Examples
Model Quality
- what is a good model?
Quality assessment approaches

- Quality of end-product depends to a great extent on the quality of early models (conceptual models, requirements)

- Most quality frameworks appear as lists of desired properties of the considered systems

- Our framework distinguishes between goals and means, and is based on linguistic concepts
ne common quality properties of information system specifications

appropriate - ability to capture germane concepts
complete - everything the software shall do is included in the specification
conceptually clean - simplicity, clarity, ease of understanding
consistent - no requirement is in conflict with other requirements
constructable - there is a systematic approach to formulating specifications
executable - automatic execution of specifications
expressive - everything relevant may be expressed without too much effort
formal - specification in formal language permitting formal analysis of spec.
precise - can find whether some realization does not meet some requirement
testable
traceable - the origin of each requirement statement is clear
unambiguous, understandable, verifiable, minimal, modifiable,…..
Weaknesses of the “list-approach” to quality assessment

- Many definitions are vague, complicated, undefined
- List is unstructured, and properties overlap
- Specification properties, language properties, and model properties are intermixed
- Some properties are design/implementation directed
- Some goals are unrealistic/impossible to reach, e.g., completeness
The framework for model quality

Domain

Model

Modeling language

Actors

Pragmatic quality

Semantic quality

Syntactic quality

M

A
Quality statements

- *The language* L consists of all of the statements that can be made according to the syntax.

- *The domain* D consists of all possible statements that are both correct and relevant for the problem at hand.

- *The model* M is the set of statements actually made.

- *The audience interpretation* A is the set of statements that the audience thinks the model contains.
Quality Goals

Syntactic quality:
Goal: "That the model is correct wrt. to the syntax of the modeling language"
Statement: \( M \setminus L = ? \)

Semantic quality:
Goal: "The model shall contain a complete and correct representation of all relevant statements from the domain"
Statement: Correct: \( M \setminus D = ? \), Complete: \( D \setminus M = ? \)

Pragmatic quality:
Goal: "The model is understood by all relevant actors. An individual actor must have understood the parts of the model relevant to him"
Statement: \( \forall i, \exists M_i = A_i \)
Quality types and means
Types of quality - more specific

- Semantic quality
- Pragmatic quality
- Syntactic quality

Diagram:

- Domain
- Model
- Tool
- Language
- Actor

Connections:

- Domain to Model
- Model to Tool
- Tool to Language
- Language to Domain
- Model to Actor
- Actor to Language

Quality:

- Semantic quality
- Pragmatic quality
- Syntactic quality
## Means for increased model quality

<table>
<thead>
<tr>
<th>Model Quality Goal</th>
<th>Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Syntactic</td>
<td></td>
</tr>
<tr>
<td>– Correct</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Formal syntax of modelling language</td>
</tr>
<tr>
<td></td>
<td>Error prevention</td>
</tr>
<tr>
<td></td>
<td>Error detection</td>
</tr>
<tr>
<td></td>
<td>Error correction</td>
</tr>
<tr>
<td>• Semantic</td>
<td></td>
</tr>
<tr>
<td>– Correct</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Statement insertion / deletion</td>
</tr>
<tr>
<td></td>
<td>Formal Semantics</td>
</tr>
<tr>
<td>– Complete</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consistency-testing</td>
</tr>
<tr>
<td>• Pragmatic</td>
<td></td>
</tr>
<tr>
<td>– Understood</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inspection</td>
</tr>
<tr>
<td></td>
<td>Aesthetics, Readability</td>
</tr>
<tr>
<td></td>
<td>Visualisation, Filtering</td>
</tr>
<tr>
<td></td>
<td>Explanation-generation, Paraphrasing, Translation</td>
</tr>
<tr>
<td></td>
<td>Execution, Animation, Simulation</td>
</tr>
<tr>
<td>• Tool support</td>
<td>Can the model be supported by proper tools and techniques?</td>
</tr>
</tbody>
</table>
SE tools

Meta-Model:
definition of model languages and relations between different languages and perspectives.

Repository:
Persistent model storage
multi-user support

Tools and techniques for improved model usage:
- Syntax testing
- Consistency checking
- Simulation
- Inspection
- Explanation generation
- Execution
- Prototyping
- …
Hvorfor var ordren ugyldig?

Flyten av ordre ble sendt av P1 fordi Kundedata ikke var gyldige.
Constructivity in DFD
Constructivity in PrM
Syntax checking

\( \text{a priori rules: never allowed} \)

\[
\text{inn-order} \rightarrow \text{ut-order}
\]

\( \text{a posteriori rules: Not allowed in a finished model} \)

Check-list:
- Name
- Flows
- Triggers/Termination
- Ports
- Flow content definitions
- Process specifications
- Data store content definitions
**Syntax explanation**

This is a process symbol. A process dynamic concept that reads a set of flows and generates a set of output-processes must have a name and an ID.

**Error explanation**

The diagram is illegal since P1 does not generate any output. All processes must generate at least one output-flow.

A data-flow cannot be directly connected between two datastores.
Dynamic Explanation (in simulation)

Why was the order rejected?

- The flow "Illegal order" was sent because the input flow customer data was not legal.

Why was customer data illegal?

- Customer data was illegal since the customer have less than €500,- on her account. This violates R3: "Orders from direct customers can only be accepted if customer has more than €500,- in balance."
Models and modeling languages are subject to variation - "as needed"

- Often tailored for specific purposes and adjusted for proper tool-support

A good model?

- It depends … on
- the Domain (D), the Language (L), the Actors (A)
  - Syntactic quality
  - Semantic quality
  - Pragmatic quality

Proper tool support provides means for increased model quality