Quality of modeling languages

John Krogstie  
Professor, IDI, NTNU  
Senior Researcher, SINTEF ICT

Overview of presentation

- Brief review of the quality framework  
- Quality of modeling languages as means for creating good models  
- Based on  
  - chapter 3.2 and 3.5 in book, material further based on  
Overall structure of framework

- Quality type (physical, empirical...)
  - One or more quality goals per quality type
    - Means to achieve this goal
      - Beneficial existing quality
      - Model properties
      - Language properties
      - Modeling activities
      - Tool-support
Why focus on language quality?

- A modeling language is a mean to be able to express knowledge in a model
- A modeling language is meant to help focusing on the important aspects of a situation
- The (modeling) language you use forms your perception of reality

- It is possible to make good models in a poor language
- It is possible to make poor models in a good language.
- It will always be possible to identify problems in any language/tool

SEQUAL – language quality

- Goals of modelling
- Modeling domain
- Model externalization
- Language extension

- Social actor explicit knowledge
- Participant appropriateness
- Modeller appropriateness
- Organizational appropriateness
- Comprehensibility appropriateness
- Technical actor interpretation
- Tool appropriateness
Aspects of language quality

- Is the language appropriate to the domain?
- Is the language appropriate for the participants knowledge of modelling languages?
- Is the language appropriate to express the knowledge of the modeller?
- Is the language appropriate for making comprehensible model?
- Is the language appropriate for tools?
- Is the language appropriate for the chosen/standardised tools and modelling languages within the organization

- Differentiate criteria for the conceptual basis of the language (language-model/meta-model), and the external representation of the language (notation)

Domain appropriateness

- $D \cap L = \emptyset$
- Basis
  - Must be able to express anything in the domain
    - Can be based on ontological analysis
    - Can be based on state of the art and state of practice in the area
  - Should not be able to express concepts not in the domain
- Notation: Must be able to express everything in the basis in a distinguishable way.
- Domain appropriateness for a general domain = expressiveness
- Support achievement of semantic quality.
Expressiveness of languages based on ontological analysis

- Bunge-Wand-Weber (BWW) as an example
- Representation model as a basis for evaluating IS analysis and design languages relative to their ability to create models that are a proper representation of the world.
- Based on Mario Bunge’s ontology
  - Well developed
  - Formalised
- BWW do not discuss if one is modelling an objective reality or a socially constructed one

Representation model as a basis for language quality

- Ontological completeness (vs. domain appropriateness)
- Ontological clarity
  - Construct overload (Comprehensibility appropriateness)
  - Construct redundancy (Comprehensibility appropriateness)
  - Construct excess (vs. domain appropriateness)
Ontological completeness

Construct excess
Expressiveness based on what is found useful within the domain of modeling

- Structural
- Functional
- Behavioral
- Rule-oriented
- Object-oriented
- Language acts
- Actor/roles-oriented

Participant appropriateness

- The stakeholders must be able to use (understand) the language efficiently
- Basis: Should be according to how people perceive reality. No standard answer on this.
  - Can judge based on general experiences from learning of modeling languages
  - Clarify the participants experience with modeling languages (and adapt if possible the languages and views to models for this)
- External representation
  - Intuitive
- Support the achievement of pragmatic quality (relative to comprehension)
Examples of intuitive symbols

- Venn-diagrams

- Amoeba-shapes or wiggly lines to indicate unfinished parts

Modeler appropriateness

- \( K \cap I^L = \emptyset \)
- Is it possible to capture work in a model, or will it always be a post-work rationalization?
- Related to the importance of tacit knowledge
- Use of metaphors etc. ala Nonaka to support making tacit knowledge explicit
- Need to be able to model vague knowledge
- Support the achievement of semantic quality
Comprehensibility appropriateness

- $L \cap \bar{L} = \emptyset$

- Support achievement of empirical and through this pragmatic quality

Aspect of comprehensibility appropriateness....

- **Basis**
  - The language is well-defined
  - Easy to differentiate between different parts of the language
  - Limited number of building blocks
  - Possible to relate concepts freely $\rightarrow$ new semantics through composition
  - Hierarchically structured
  - Uniform use of building block
  - Flexibility in detail
  - Expressive economy:
    - Frequent and important statements can be expressed concisely
Construct overload

Ontological concepts

Design-concepts

Construct redundancy

Ontological concepts

Design-concepts
Improvement of comprehension continues...

- **External representation**
  - Easy to differentiate different symbols
  - Easy to see which symbol any mark is part
  - Uniform use of symbols
  - Symbolic simplicity
  - Use of color
  - Emphasis of what is important
    - Symbol size, solidity, color, differences, movement, number of edges.
  - Support in making esthetical models
  - Special symbol for important concepts and statements

Different ways of illustrating associations between classes (but better than predecessor)
Mix of pictographic and logographic symbols in use case modeling

Example of misguided emphasis
Improvement of technical actor interpretation

- The language is appropriate for automatic reasoning and execution.
- Formal syntax
- Formal semantics (operational and/or mathematical)
  - Discover inconsistencies and ambiguity
  - Formal proofs possible
  - Translation into executable models
  - Ensure that different participants interpret the model in the same way.
- Analyzability/executability
- Support the achievement of syntactical, semantic and pragmatic quality.

Organizational appropriateness

- Language standardization (independent of tools)
  - Increase the potential for reuse
  - Support organizational learning
  - Enable a common methodology
- Tool standardization
  - Decrease cost of modeling tools
  - Easier to transfer models between different modeling task (within a project) and modeling activities (between projects)
Trade-off between different criteria

- The different criteria influence each other, and partly works against each other.
  - Intuitivity vs. the number of symbols: Chinese vs. English
  - Intuitivity vs. Simplicity of symbols: Classes vs. instances in ERAE
  - Expressiveness vs. Symbol simplicity: EEML vs. DFD
  - Expressive economy vs. Number of symbols: EEML – bidirectional arrows

- A perfect language is not possible to make, weaknesses in a language can be made less acute through method and tool support.

Language representation as a model

- A language model typically contains:
  - Meta-model (for language semantic)
    - Conceptual meta-model (for human understanding)
    - Logical meta-model (e.g. class diagrams for logical consistency)
    - Physical meta-model (for tool developers)
  - Notation (for syntax and visual appearance)

- Might also be in the form of tutorials etc.

- How a language is supported in a tool can also be evaluated using the quality levels.
Quality of a language model

- **Physical**
  - The language model is available and can be updated if appropriate.

- **Empirical**
  - As other models

- **Syntactic**
  - All examples follow the syntax of the language. The meta-model follows the syntax of the meta-modeling language

- **Semantic**
  - The whole language is described correctly. The meta-model and the notation guide is consistent.

- **Pragmatic**
  - Indexes, cross-references, tutorials,

- **Social**
  - vs. Language standardization, and agreement related to meta-model extensions

- **Organizational**
  - Minimize training time

---

Coming up

- **Wednesday**
  - Quality of UML: Analytical evaluation
  - Quality of process modeling languages: Comparing EEML, BPMN, and UML activity diagrams for process modeling relative to introducing SOA in an organization: Combined Analytical/Empirical evaluation
Quality of modeling languages

John Krogstie
Professor, IDI, NTNU
Senior Researcher, SINTEF ICT