Quality of models

John Krogstie
Professor, IDI, NTNU
Overview of presentation

- What do we mean with ‘quality’?
- Presentation of semiotic model quality framework (SEQUAL)
Different views on quality

- Product vs. Process quality (e.g. CMM)
- A good model feels right and does not appear to have extraneous detail (Rumbaugh 1991)
- "The quality that can be defined is not true quality" - Robert Pirsig
- According to requirements (ISO 9000)
- The user is satisfied (Denning)
- Kano-model: Normal, exciting and expected requirements
- Properties of the product (-ilities) ISO-IEC 9126
- Properties of a requirements specification or model (Davis/Pohl)
- Data and data model quality (Batini/Shanks)
- Quality related to different semiotic levels (Lindland/Stamper)
ISO 9000 view of quality

- Traditional:
  - a product has high quality if it is according to its specification

- ISO9000-2005:
  - quality is the “Degree to which a set of inherent characteristic fulfils requirements."
  - ‘requirement‘ is defined as needs or expectations from a customer (and no longer as necessarily explicit specifications of such needs)
Denning (1992) – 3 levels of (software) quality

- 1. level: a product is in ‘accordance to the specification’
- 2. level: there are no negative side-effects of the installed information system
- 3. level: the information system enables additional information system support to its users not conceived in the first place, i.e. actually giving the users *more* of what they need than what was promised in the specification
Kano-model of requirements

- User Satisfaction
- User Dissatisfaction
- Fulfills Expectations
- Fulfills Expectations
- Normal Requirements
- Expected Requirements
- Exciting Requirements
Quality characteristics of software products (ISO/IEC 9126)

- Functionality: Does the software support all the required functions?
- Reliability: How reliable is the software?
- Efficiency: How efficiently does the software perform?
- Usability: How easy is the software to use?
- Portability: How easy is it to transfer the software to another (technical) environment?
- Maintainability: How easy is the software to modify?

- Sub-characteristics (24)
- Metrics (113)
Moody/Shanks– Quality of data model

- **Correctness** is defined as whether the model conforms to the rules of the data modelling technique. This includes diagramming conventions, naming rules, definition rules, rules of composition and normalisation.

- **Completeness** refers to whether the data model contains all information required to support the required functionality of the system.

- **Integrity** is defined as whether the data model defines all business rules that apply to the data.

- **Flexibility** is defined as the ease with which the data model can cope with business and/or regulatory change.

- **Understandability** is defined as the ease with which the concepts and structures in the data model can be understood.

- **Simplicity** means that the data model contains the minimum possible entities and relationships.

- **Integration** is defined as the consistency of the data model with the rest of the organisation’s data.

- **Implementability** is defined as the ease with which the data model can be implemented within the time, budget and technology constraints of the project.
Quality characteristics of a requirements specification (Davis)

- Unambiguous
- Complete
- Correct
- Understandable
- Verifiable
- Internally Consistent
- Externally Consistent
- Achievable
- Concise
- Design-independent
- Traceable
- Modifiable
- Electronically Stored
- Executable/Interpretable/Prototypable
- Annotated by Relative Importance
- Annotated by Relative Stability
- Annotated by version
- Not Redundant
- At Right Level of Detail
- Precise
- Reusable
- Traced
- Organised
- Cross-referenced
Dimensions related to requirements specification process (Pohl)
Core framework on quality of models (1994)

- Domain
- Language
- Model
- Audience interpretation

Quality dimensions:
- Appropriateness
- Semantic quality
- Syntactic quality
- Pragmatic quality

Connections:
- Domain to Language: Appropriateness
- Domain to Model: Appropriateness
- Model to Audience interpretation: Pragmatic quality
- Language to Model: Syntactic quality
Some weaknesses with earlier work on quality of models

- Only for some types of models (Requirements specification/design/completed product)
- Cover only models in some languages
- Uncoordinated list of wanted properties
- Mix quality of process, tool, language, and model
- Mix quality goals and means to achieve these goals
- Is better suited for limited, constructed cases than real modelling problems in practice.
SEQUAL – A framework for understanding and assessing quality of models based on semiotics

- For models as a knowledge representation in general
- Can be extended and specialised towards specific types of model and modelling languages
- Differentiate between goals of modelling (quality characteristics) and means to achieve these goals
- Set-oriented definition to enable a formal discussion of the different quality levels
- Differentiate between quality of different levels based on semiotic theory
- Takes into account that models are socially constructed
Main elements of a modelling activity

Example of goal: Create a requirements specification for a travel agency on the net.
Sets in the quality framework

- **A**: Actors that develops or has to relate to (parts of) the model. Can be persons or tools (technical actors).

- **L**: What can be expressed in the modelling language

- **M**: What is expressed in the model

- **D**: What can be expressed about the domain (area of interest)

- **K**: The explicit knowledge of the participating persons

- **I**: What the persons in the audience interpret the model to express

- **T**: What relevant tools interpret the model to say

- **G**: The goals of the modelling

All of these sets evolves as part of modelling
Usage of modeling and models

1. Sensemaking
2. Communication
3. Computer-assisted analysis
4. Quality assurance
5. Model deployment
6. Context for change

Current state

Future state

Model of current state

Model of future state
Different types of goals possible - example

- **Communication and sense-making around models of the current state**
  - The models developed should help sharing best practice between different units of the organisation.
  - The models developed should be helpful in the refining of the processes.

- **Communication around models of the future state**
  - The new work process should be documented through the models.
  - The models developed should help harmonise the current work processes across different parts of the organisation.
  - The models developed should be used to teach the software developers about the domain.

- **Computer-assisted analysis**
  - The models developed should help analyse the current work processes

- **Model deployment**
  - The models developed should be used as a procedural tool in everyday work.
  - The models developed should support the use of the software application developed for process-support.

- **Context for change**
  - The models developed should define the scope of the software application
Who are the human actors (A)?

- A subgroup of those that are influenced by the modelling (stakeholders).
- Different types of stakeholders. E.g. in the context of system development:
  - System analysts, designers and programmers
  - Users
  - Managers
What is expressed in a model?

- This is based on the model and the semantics of the language used for modelling.
- Can differentiate between what is explicitly and implicitly expressed.
  - Language: Propositional logic
  - Explicit model: A, A -> B
  - Implicit model: B
  - Overall model: A, B, A -> B
Different types of domains relevant for IS-development

- EIS: Existing Information System
- ECIS: Existing Computer-based Information System
- FIS: Future Information System
- FCIS: Future Computer-based Information System
  - Requirements specification: Looking at the externally observable behaviour of the future CIS
  - Design: Looking at the internal fabric of the future technical system
- CIS: The actual computerised information system, including the data within the system.
- In general, what is part of the domain is dependent on the goal of modelling
Different domains for IS-development according to MDA-thinking in OMG

- CIM (vs. M(FIS)): Computational Independant Model
- PIM (vs. M(FCIS) - RE/Design): Platform Independant Model
- PSM (vs. M(FCIS) – Detailed design): Platform Specific Model
SEQUAL

Goals of modelling

Modeling Domain

Model externalization

Social actor interpretation

Language extension

Social actor explicit knowledge

Perceived semantic

Social

Deontic

Physical

Pragmatic (human understanding)

Syntactic

Deontic

Semantic

Empirical

Pragmatic (tool understanding)

Technical actor interpretation

Deontic (learning)

Deontic (action)
Overall structure of framework

- Quality type (physical, empirical...)
  - One or more quality characteristics per quality type
    - Means to achieve the quality characteristics
      - Beneficial existing quality
      - Model properties
      - Language properties
      - Modeling activities
      - Tool-support
Model example to illustrate the different quality levels

- Language: ER
- Domain: Conference organizing
- Goal: Design of database solution to support conference organizing
Physical Quality

- Internalizability
  - Model persistence
  - Model availability
  - Currency
  - Security
-> Database functionality (model repository)
Empirical quality

- Look on aspects related to
  - Ergonomics
  - Graph and document layout
  - Readability

- The model must be externalised (physical quality)

- Language properties
  - Expressive economy
  - Use of emphasis including colour

- Modelling and tool activities
  - (Automatic) graph-layout, readability index, grammar checking, evaluation of use of colour.
Example of poor graph-layout
Syntactic quality

- Syntactic correctness: $M \setminus L = \emptyset$
- Two types of errors
  - Syntactic invalidity
  - Syntactic incompleteness
- The model must be externalised (physical quality)
- Language properties
  - Formal syntax
- Activities
  - Error prevention
  - Error detection
  - Error correction (automatically or by suggestion ("spellcheck") )
Example of syntactic invalidity
Example of syntactic incompleteness
Semantic quality

- Quality characteristics
  - Validity: $M \setminus D = \emptyset$
  - Completeness: $D \setminus M = \emptyset$

- Necessary/useful that the model is externalised and is syntactically correct

- Language properties: Formal semantics

- Activities: Model testing (consistency checking), reuse of models, ’driving questions’, meta-model adaptation
Example of semantic invalidity (and incompleteness)
Pragmatic quality

- Quality characteristics
  - Comprehension, do the audience understand what the model express? (I=M)

- Useful that the model have high physical, empirical, and syntactic quality before evaluating pragmatic quality.

- Language properties:
  - Operational semantics
  - Executability
  - Explicit modelling of intention

- Activities: Inspection, visualization, filtering/views, explanation generation, simulation, animation, reporting, execution/prototyping, model-generated solutions
Example on language-oriented view

![Diagram showing the relationship between Person, Author-ship, and Paper with cardinalities N and M.](image-url)
Example of model-oriented filter

- Paper
  - Title
  - Language
Example of model translation
Perceived semantic quality

- Quality characteristics
  - Perceived validity $\mathbf{I\setminus K = \emptyset}$
  - Perceived completeness: $\mathbf{K\setminus I = \emptyset}$

- Useful that the model has high physical, empirical, syntactic, and pragmatic quality before investigating perceived semantic quality

- Same means and activities as for semantic quality.
Social quality

- Quality characteristics: Agreement
  - Agreement in knowledge/interpretation/model
  - Relative vs. absolute agreement
- Important first to address physical, pragmatic and perceived semantic quality
- Language properties: Possibility to explicitly express inconsistencies based on disagreement.
- Activities: Model integration and conflict resolution
Deontic quality

- The deontic quality of the model relates to
  - that all statements in the model contribute to fulfilling the goals of modelling (goal validity)
  - that all the goals of modelling are addressed through the model (goal completeness)
- Deontic quality introduce a context that relax wanted quality for a model on the other levels (e.g. trade-off between completeness of the model relative to cost).
- Expressed with the notion of feasible quality (particularly on the levels of semantic, pragmatic, perceived semantic and social quality)
- Goals include also aspects relative to participant learning and domain improvement
Model-based Development and Evolution of Information Systems

Physical vs social aspects

Goals of Modelling G

Physical

Modeling Domain D

Language Extension L

Model externalization M

Social

Stakeholder explicit knowledge K

Modeller explicit knowledge K

Social actor Interpretation I

Technical actor Interpretation T

Knowledge appropriateness

Participant appropriateness

Modeller appropriateness

Comprehensibility appropriateness

Physical appropriateness

Social pragmatic

Tool appropriateness

Tool comprehension support

Perceived Semantic

Semantic

Syntactic

Empirical

Technical pragmatic

Domain appropriateness

Participant appropriateness

Modeller appropriateness

Comprehensibility appropriateness

Physical appropriateness

Social pragmatic

Tool appropriateness

Tool comprehension support

Social

Physical Social

Syntactic

Semantic

Empirical

Technical pragmatic

Domain appropriateness

Participant appropriateness

Modeller appropriateness

Comprehensibility appropriateness

Physical appropriateness

Social pragmatic

Tool appropriateness

Tool comprehension support

Social
SEQUAL – language quality

Goals of modelling

Modeling domain

Model externalization

Social actor explicit knowledge

Modeller appropriateness

Organizational appropriateness

Participant appropriateness

Comprehensibility appropriateness

Language extension

Social actor interpretation

Tool appropriateness

Technical actor interpretation

Domain 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 proving tool support necessary to reach the goals of modelling?
- 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)
SEQUAL specializations
Usage of the framework

- E.g. in ATHENA (EU project)
  - Evaluation of a modeling language under development
  - Evaluation of the model of the modeling language (meta-model)
  - Evaluation of a modeling tool/environment
  - Evaluation of a modeling methodology
    - The methodology as a model
    - The way the methodology support development of models of high quality

- Evaluation and choice of modeling languages (UML, BPMN, EEML, others)

- Evaluation of models

- Methodology guidelines for developing good models

- Guidelines for developing new modeling languages
Quality of models

John Krogstie
Professor, IDI, NTNU