

Quality of Conceptual Data Models¹

John Krogstie

IDI, NTNU, Høgskoleringen 1 N-7491, Trondheim, Norway
krogstie@idi.ntnu.no

Keywords: Semiotic quality model: data model

Abstract: Since the introduction of the ER-language in the late seventies, data modelling has been an important area in information systems development. Data modelling both on the conceptual and logical level is still widely used. The quality of data models have also been investigated and discussed since the mid-nineties with work by among others Batini and Moody and Shanks. In this paper we present a specialization of a general framework for assessing quality of models based on organizational semiotics for being able to evaluate the quality of conceptual data models. Comparing the approaches we find on the one hand that the described properties of data model quality is subsumed by the semiotic framework on a high level, and that there are aspects in this framework that are not covered by the existing work on data model quality. On the other hand, the comparison has resulted in a useful deepening of the generic SEQUAL-framework, and in this way improved the practical applicability of SEQUAL when applied to discussing the quality of conceptual data models.

¹ This article was presented at ICISO 2013 <http://www.orgsem.org/2013/>

1 INTRODUCTION

A conceptual model is traditionally defined as a description of the phenomena in a domain at some level of abstraction, which is expressed in a semi-formal or formal visual language (Krogstie, 2012). The field has its roots in information systems development and computer science with methodologies like Data Flow Diagram (DFD) (Gane and Sarsons, 1979), Entity Relationship diagrams (ER) (Chen, 1976) and more recently Unified Modeling Language (UML) (Booch, Rumbaugh and Jacobson, 2005), goal-oriented models (e.g. GRL/i* (URN, 2012)) and Business Process Modeling Notation (BPMN) (OMG, 2011). Although a number of newer approaches to modelling have appeared, data modelling in the form of e.g. ER-modelling is still quite popular in practice (Davies et al., 2006), which makes it important to have an understanding on the quality of data models.

Since the early nineties, much work has been done relative to analyzing the quality of models. Early proposals for quality goals for conceptual models and requirement specifications as summarized by (Davis et al., 1993) included many useful aspects, but unfortunately as a poorly structured list mixing quality of models, modelling languages, modelling methods and modelling tools. They are also often restricted in the kind of models they cover (Moody, 2005) (e.g. requirements specifications (Davis et al., 1993)) or the modelling language (e.g. ER-models (Moody and Shanks, 1994) or process models (Hepp and Roman, 2007; Sedera et al., 2005)). More comprehensive and generic framework for evaluating modelling approaches has been developed (Krogstie and Lillehagen, 2008; Nelson et al., 2011), but these again can become overly general for practical use.

Inspired by (Moody, 2005), suggesting the need for an inheritance hierarchy of quality frameworks, we will in this paper provide a specialization of the generic SEQUAL framework (Krogstie, 2012) for the evaluation of the quality of conceptual data models.

In section 2, we present overall goals of modelling in general and data modelling in particular. Existing work on quality of data models is described in section 3. Section 4 provides a brief overview of SEQUAL, whereas the specialization of SEQUAL for quality of data models is described in section 5. The main research questions that we try to address with this work are:

- Is it possible to specialise the SEQUAL framework covering current aspects of data model quality as described in the literature?
- Will this specialization extend and introduce new areas of concern for data model quality?

Section 6 consists of a short conclusion and overview of planned work for developing and further evaluating the proposed approach.

2 GOALS AND LEVELS OF DATA MODELING

Conceptual modelling including data modelling is usually done in some organizational setting. One can look upon an organization and its information system abstractly to be in a certain state (the current state, often represented as a descriptive 'as-is' model) that are to be evolved to some future wanted state (often represented as a prescriptive 'to be' model). Obviously, changes will happen in an organization no matter what is actually planned, thus one might in practice have the use for many different models and scenarios of possible future states, but we simplify the number of possible future states in the discussion below.

The state of the organization includes the existing processes, organizational structures and computer systems. These states are often modelled, and the state of the organization is perceived (differently) by different persons through these models. This opens up for different usage areas of conceptual models as described in (Krogstie, 2012):

1. Human sense-making: The model of the current state can be useful for people to make sense of and learn about the current situation as it is perceived.
2. Communication between people in the organization: Models can have an important role in human communication. Thus, in addition to support the sense-making process for the individual, a model can act as a common framework supporting communication between different people.
3. Computer-assisted analysis: To gain knowledge about the organization through simulation or deduction, often by comparing a model of the current state and a model of a future, hopefully better state.

4. Quality assurance, ensuring e.g. that the organization acts according to a certified process decided for instance through an ISO-certification process.
5. Model deployment and activation: To integrate the model of the future state in an information system directly. Models can be activated in three different ways:
 - i. Through people, where the system offers no active support.
 - ii. Automatically, where the system plays an active role, as in an automated workflow system.
 - iii. Interactively, where the computer and the users co-operate to bring the process forward.
6. To give the context for a traditional system development project, without being directly activated.

Data models are a type of structural models used both for human sense-making and communication (areas 1 and 2 above) and as a context for systems development (area 6 above). Approaches within the structural modelling perspective concentrate on describing the static structure of a domain. The main construct of such languages is the "entity". Other terms used for this with some variations in semantics are object, concept, thing, and phenomena. Going back to the ANSI SPARC work (Tsichritzis and Klug, 1978), one differentiates between three levels of data models:

- Conceptual models (e.g. ER models)
- Logical models (e.g. in the form of relational tables)
- Physical models (e.g. a physical implementation of a relational database using a particular DBMS e.g. Oracle)

There exist well-defined ways of transforming models between these levels, although often automatic mappings are not sufficient in practice to get ideal database performance based on the conceptual and logical models.

When working with conceptual models, we typically concentrate on the conceptual level, although often with the goal of producing logical/physical models to be included as part of running information systems. Thus we will focus on quality of conceptual data models in this article.

3 EXISTING WORK ON QUALITY OF DATA MODELS

Some of the early work on quality of models focused on data models (Moody and Shanks, 1994), work that was extended in (Moody, 1998; Moody and Shanks, 2003) based on empirical investigations on its use. This body of work is together with the work of (Batini and Scannapieco, 2006) described below to our knowledge the most widely cited in this area. Moody and Shanks (2003) contains the following wanted characteristics and metrics for data model quality:

Correctness is defined as whether the model conforms to the rules of the data modelling technique (i.e. whether it is a valid data model). This includes diagramming conventions, naming rules, definition rules, rules of composition and normalisation. Proposed metrics:

1. Number of violations to data modelling standards
2. Number of instances of entity redundancy
3. Number of instances of relationship redundancy
4. Number of instances of attribute redundancy

Completeness refers to whether the data model contains all information required to support the required functionality of the system. Proposed metrics:

5. Number of missing requirements
6. Number of superfluous requirements
7. Number of inaccurately defined requirements
8. Number of inconsistencies with the process model

Integrity is defined as whether the data model defines all business rules that apply to the data. Proposed metrics:

9. Number of missing business rules
10. Number of incorrect business rules
11. Number of business rules inconsistent with the process model
12. Number of business rules redundantly defined in process model rules

Flexibility is defined as the ease with which the data model can cope with business and/or regulatory change. Proposed metrics:

13. Number of data model elements which are subject to change
14. Probability adjusted cost of change
15. Strategic impact of change

Understandability is defined as the ease with which the concepts and structures in the data model can be understood. Proposed metrics:

16. User rating of understandability
17. User interpretation errors
18. Application developer rating of understandability
19. Subject area-entity ratio
20. Entity-attribute ratio

Simplicity means that the data model contains the minimum possible entities and relationships still representing the domain. Proposed metrics:

21. Number of entities (E)
22. System complexity (E+R)
23. Total complexity (aE+bR+cA)

Integration is defined as the consistency of the data model with the rest of the organisation's data. Proposed metrics:

24. Number of data conflicts with a Corporate Data Model
25. Number of data conflicts with existing systems
26. Number of data items duplicated in existing systems or projects
27. Rating of ability to meet corporate needs

Implementability is defined as the ease with which the data model can be implemented within the time, budget and technology constraints of the project. Proposed metrics:

28. Development cost estimate
29. Technical risk rating

Based on an empirical investigation (Moody and Shanks, 2003) (which mainly perceived only metric 22, 26, and 28 to be cost beneficial to keep track of in the context of the particular case), two additional metrics were proposed

Metric 30. Reuse Level. This is the inverse or "positive" of the level of duplication metric (Metric 26) and measures the number of existing data items reused as part of the new model

Metric 31. Number of Issues by Quality Factor. Each quality issue raised as a result of quality reviews can be classified by the quality factor it relates to. The number of issues raised and their severity by quality factor gives a "defect frequency" which can be used for comparison over time

Although one learning from (Moody and Shanks, 2003) is that one will want to limit the number of metrics in a particular project, it is not from this given what metrics to include in all projects, thus we aim below for a more complete overview of aspects of data model quality, of which not necessarily all are relevant to use in all projects.

Another much cited overview of data model (schema) quality is presented in (Batini and Scannapieco, 2006), containing the following quality characteristics:

- Correctness with respect to the model concerns the correct use of the concepts in the language. The negative example is to represent FirstName as an entity and not as an attribute (since FirstName can be argued to not have unique existence in the real world).
- Correctness with respect to requirements
- Minimalisation, no requirement is represented more than once
- Completeness
- Pertinence that measures how many unnecessary conceptual elements are included
- Readability through aesthetics
- Readability through simplicity
- Normalisation

Whereas one can argue that the last applies first on the logical level, the others apply on the conceptual level and will be included below where we discuss each quality level in detail in SEQUAL, starting with those areas that are specifically discussed by Moody/Shanks and Batini. The aspects from the work of Moody and Shanks are highlighted with a starting M-, and those from Batini are represented with a starting B- when positioning them within SEQUAL in section 5. First we present the generic SEQUAL framework.

4 INTRODUCTION TO SEQUAL

SEQUAL has the following properties (Krogstie, 2012):

- It distinguishes between goals and means by separating what you are trying to achieve (quality of models) from how to achieve it.
- It can be used for evaluation of models and modelling languages in general, but can also be extended for the evaluation of particular types of models as we will see an example of in the next section.
- It is closely linked to linguistics and semiotics. In particular, the core of the framework including the discussion on syntax, semantics, and pragmatics is parallel to the use of these notions in the semiotic theory of Morris (see e.g. (Nöth, 1990) for a brief introduction). Extensions are partly based on extensions in organizational semiotics (Falkenberg et al., 1996), using the levels physical, empirical, syntactic, semantic, pragmatic, and social as in the semiotic ladder of Stamper. We have retained the original terminology from these areas.
- It is based on a constructivistic world-view, recognizing that models are usually created as part of a dialogue between the participants involved in modelling, whose knowledge of the modelling domain and potentially the domain itself changes as modelling takes place.

The framework has earlier been used for evaluation of modelling and modelling languages of a large number of perspectives, including object (Krogstie, 2003), process (Recker et al, 2007), enterprise (Krogstie and Arnesen, 2004), and goal-oriented (Krogstie, 1999; Krogstie, 2008) modelling. Quality has been defined referring to the correspondence between statements belonging to the following sets illustrated in Fig. 1:

- G , the set of goals of the modelling task.
- L , the language extension, i.e., the set of all statements that are possible to make according to the rules of the modelling languages used.
- D , the domain, i.e., the set of all statements that can be stated about the situation.
- M , the externalized model itself.
- K , the explicit knowledge of the audience relevant to the domain.
- I , the social actor interpretation, i.e., the set of all statements that the audience interprets that an externalized model consists of.
- T , the technical actor interpretation, i.e., the statements in the model as 'interpreted' by modelling tools.

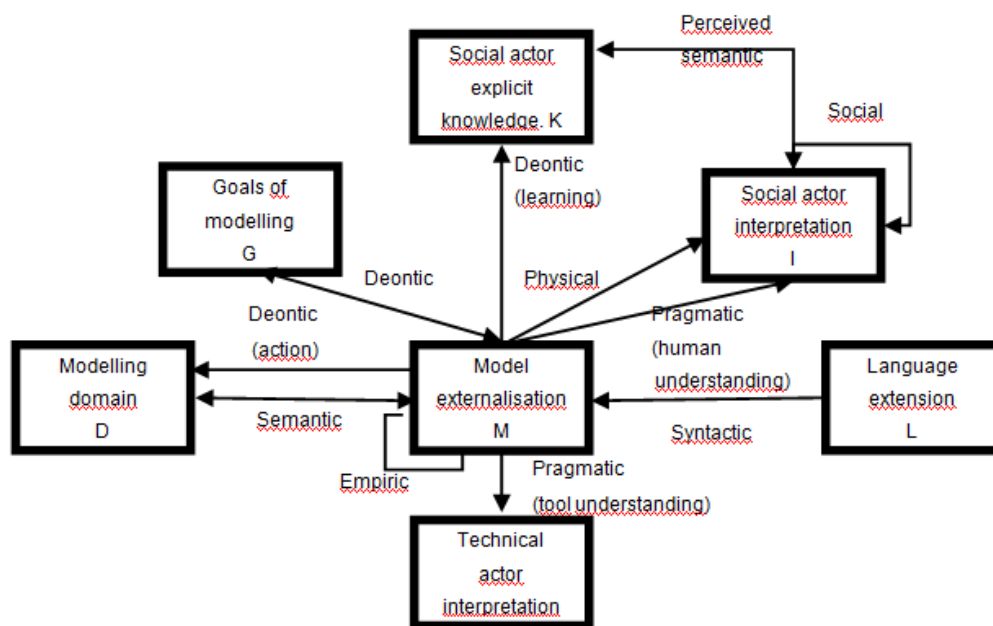


Figure 1: SEQUAL framework for discussing quality of models

Quality is defined according to the following levels:

- Physical quality: The basic quality goal is that the externalized model M is available to the relevant social and technical actors.

- Empirical quality deals with comprehension and predictable error frequencies when a model M is read or written by different social actors
- Syntactic quality is the correspondence between the model M and the language extension L .
- Semantic quality is the correspondence between the model M and the domain D . This includes validity and completeness. Domains can be divided into two parts, exemplified by looking at a software requirements specification (Davis et al., 1993). Everything the computerized information system is supposed to do (for the moment ignoring the different views the stakeholders have on the CIS to be produced) is termed the *primary domain*. Constraints on the model because of earlier baselined models such as system level requirements specifications, enterprise architecture models, statements of work, and earlier versions of the requirement specification to which the new requirement specification model must be compatible is termed the *modelling context*.
- Perceived semantic quality is the similar correspondence between the social actor interpretation I of a model M and his or hers current knowledge K of domain D .
- Pragmatic quality is the correspondence between the model M and the actor interpretation (I and T) and application of it. One differentiates between social pragmatic quality (to what extent people understand and are able to learn from and use the models) and technical pragmatic quality (to what extent tools can be made that can interpret the models).
- The goal defined for social quality is agreement among social actor's interpretations.
- The deontic quality of the model relates to that all statements in the model M contribute to fulfilling the goals of modelling G , and that all the goals of modelling G are addressed through the model M . In particular, one include under deontic quality the extent that the participants after interpreting the model learn based on the model (increase K) and that the audience are able to change the domain D if this is beneficially to achieve the goals of modelling. This area was earlier called organizational quality. The term deontic is from Greek 'deon' - duty from impersonal dei - it behoves (i.e. it is fitting) relating to the goal one want to achieve.

5. QUALITY OF CONCEPTUAL DATA MODELS

In this section we specialise SEQUAL in particular taking into account the work on quality of data models described by Moody and Shanks, and Batini as outlined in section 3.

5.1 Physical Quality of a Conceptual Data Model

No measures for physical quality are included in the work of Batini and Moody. The normal measures of persistence, currency and availability applies as with all other models.

- Persistence: How persistent is the model, how protected is it against loss or damage? For a model on disk, the physical quality will be higher if there is a backup copy, even higher if this backup is on another disk whose failure is independent of the original. Similarly, for models on paper, the amount and security of backup copies can be essential. The way of storing the model should be efficient, i.e. not using more space than necessary. A simple metric for persistence is the proportion of model-statements that are electronically stored in a model repository.
- Currency: How long time ago is it since the model statements were included in the model (assuming the statements were current when entered). Depending on the type of model, the age of the model statements is of varying importance. When the domain is changing rapidly (has high volatility), currency of the stored model is of more importance for the model to have appropriate timeliness. Metrics on currency can be devised and calculated easily if the model repository support time-stamping of statements. This area will relate to semantic quality (see below), relative not only to the time of entering of a model statement, but also the last time the model statement is validated.
- Availability: How available is the model to the audience? Clearly, this is dependent on that the model is externalised and made persistent in the first place. Availability also depends on distributability, especially when members of the audience are geographically dispersed. A model which is in an electronically distributable format will be more easily distributed than one which must be printed on paper. It may also matter

exactly what is distributed, e.g. the model in an editable form or merely in an output format, or a format where you can add annotations, but not change the actual model.

- A metric for availability is the proportion of model statements relevant for a member of the audience being available for that audience member. In connection to currency and availability, the term 'timeliness' is often used, i.e. the model is not only current, but are also available in time for events that corresponds to their usage. This relates directly to the goal of modelling, thus timeliness is set up as a deontic goal below. A possible measurement of timeliness consists of (i) a currency measurement and (ii) a check that the model is available before the planned usage time.
- Security can be an issue on some models, i.e. that it is only the authorised people that have access to and can change the model.

Many of the modelling techniques and tool functionality in connection with physical quality are based on traditional database-functionality using a model-repository-solution for the internal representation of the model.

5.2 Empirical Quality of a Conceptual Data Model

The area of empirical quality is supported with some of the metrics under M-understandability (metric 19 and 20). In addition it can be argued that the metrics under M-simplicity using metric 21, 22 and 23 and B-readability through conciseness applies as means at this level (similarly as concise is a mean for empirical quality of an software requirements specification (SRS) as discussed by (Krogstie, 2001). Traditional guidelines for graph aesthetics (Battista et al., 2004; Tamassia et al., 1988) covered also by B-readability through *aesthetics* apply in the following way:

- Angles between edges going out from the same node should not be too small. An additional aspects that makes this specifically relevant for a data model is when cardinality constraints are given with annotations which you find in many data modelling languages. In such cases it is important to see what each mark is part of, so called syntactic disjointness (Goodman, 1976).
- Minimise the area occupied by the diagram.
- Balance the diagram with respect to the axis.

- Minimise the number of bends along edges in the diagram.
- Minimise the number of crossings between edges.
- Place nodes with high degree in the centre of the model. This is typically central entities, whose positioning in the middle also will help to emphasise these as more important.
- Minimise differences among nodes' dimensions (given nodes of the same type). A challenge here can specifically be in languages where attributes are written inside the entity-class symbols (e.g. UML class diagrams). A positive aspect of this type of languages is that the attributes are not represented by a separate node, thus keeping the number of nodes lower.
- Minimise the global length of edges
- Minimise the length of the longest edge.
- Have symmetry of sons in hierarchies. In particular relevant when you depict generalisation-hierarchies.
- Have uniform density of nodes in the model.
- Have verticality of hierarchical structures. This means that in a tree/hierarchy, nodes at the same level in the tree are placed along a horizontal line with a minimum distance between. Also applies in particular to structures such as generalisation and aggregation hierarchies.

One can also devise guidelines for the naming of concepts, depending a bit on the concrete language. E.g. for an ER-model one would have:

- Entity classes should be named with nouns and noun phrases in singular form. If a noun phrase is used (in English), use spaces to divide the words
- Relationship classes should be named with nouns. Note that in languages where the role-name on each side of the relationship class is represented, these should be named with verb phrases. For instance in ORM, these are mandated to be in so-called mixfix-notation, to support automatic verbalisation (Halpin and Curland, 2006) as a paraphrasing technique to support pragmatic quality (see below).
- Attributes should be named using nouns or adjectives. The names should be unique within an entity class (different entity classes can have attributes with the same name)

When developing the logical and physical data models from the conceptual models, there might be additional guidelines, some of which are technology

specific (e.g. due to reserved words in the DBMS used). It should not be necessary to worry about these kinds of aspects at the conceptual level.

5.3 Syntactic Quality of a Conceptual Data Model

Parts of M-correctness (metric 1: number of violations to data modelling standard) relates to syntactic quality. We expect that this also includes rules for the language used for describing business rules (Business rules being only concretely mentioned under the area M-integrity). From the generic SEQUAL framework we have one syntactic quality characteristic, syntactical correctness, meaning that all statements in the model are according to the syntax and vocabulary of the language.

Syntax errors are of two kinds:

- Syntactic invalidity, in which words or graphemes not part of the language are used.
- Syntactic incompleteness, in which the model lacks constructs or information to obey the language's grammar.

5.4 Semantic Quality of a Conceptual Data Model

When looking upon semantic quality relative to the primary domain of modelling, we have the following properties: M-Completeness (metric 5 number of missing requirements) and M-integrity (metric 9: number of missing business rules) relates to completeness. The same applies to Batini's point on B-completeness.

M-Completeness (metric 6: number of superfluous requirements) and M-integrity (metric 10: number of incorrect business rules) relates to validity. The same applies to Batini's points on B-correctness with respect to model and B-correctness with respect to requirements.

M-Completeness (metric 7: number of inaccurately defined requirements) relates to precision, which can be a matter of either incompleteness or invalidity (cf. (Krogstie, 2001)). Inconsistency within the data model is similarly either an example of incompleteness or invalidity. Given a parallel development of a process model, M-completeness (metric 8: number of inconsistencies with process model) and M-integrity

(metric 11: number of business rules inconsistent with the process model) also falls into this area. If the process model is rather part of the model context, it can be positioned together with other relations to the model context:

Properties related to the model context are related to the area M-integration:

- Metric 24. Number of data conflicts with Corporate Data Model
- Metric 25. Number of data conflicts with existing systems

Some additional semantic means mentioned are related to redundancy. Unlike the other properties, redundancy is not necessarily bad. Redundancy can in fact improve empirical and pragmatic quality (see below) at the cost of conciseness. This relates to M-correctness (metrics 1, 2, 3), M-integrity (metric 12), M-integration (metric 26) and B-minimalisation.

M-Reuse (metric 30) can also be looked upon as a technique that potentially improves completeness.

The main problem of redundancy appears when the data model is changed, thus redundancy is problematic relative to M-flexibility. The concrete metrics suggested for this is positioned under deontic quality.

5.5 Pragmatic Quality of a Conceptual Data Model

The following metrics under M-understandability is positioned as part of pragmatic quality (and not under empirical quality) due to the concrete mentioning of *user* ratings and *user* interpretation. This includes metric 16, 17, and 18. Verbalisation (Halpin and Curland, 2006) is an interesting technique for making it easier to understand the information captured in the data models. The verbalisation language for ORM 2 was architected to meet five main design criteria: expressiveness, clarity, flexibility, localizability, and formality.

- For reasons of expressiveness, both alethic and deontic modalities (Krogstie and Sindre, 1996) are supported in the language
- Localisation as well as support of natural verbalisation for predicates of any arity dictate the use of mixfix predicates (e.g. ... introduced ... to ... on ...)
- For clarity and flexibility reasons, constraint verbalisations may be presented in positive or negative form (showing how to satisfy or violate the constraint), and may use relational

or attribute style (employing predicate readings or role names) or a mix of the two.

5.6 Social Quality of a Conceptual Data Model

The goal defined for social quality is agreement. Six kinds of agreement have been defined, according to the following dimensions:

- Agreement in knowledge vs. agreement in model interpretation. In the case where two models are made based on the view of two different actors, we can also talk about agreement between models.
- Relative agreement vs. absolute agreement.

Relative agreement means that the various sets to be compared are consistent -- hence, there may be many statements in the model of one actor that are not present in that of another, as long as they do not contradict each other. Absolute agreement, on the other hand, means that all models are the same. In practice relative agreement is what one should strive for. Neither Moody nor Batini address this area. Practical approaches to support this relate to compare and merge different models developed by different stakeholders. Techniques for schema integration (Francalanci and Pernici, 1993) are relevant for this area. The process can be considered as consisting of four sub-processes:

1. Pre-integration: When more than two models are used as input to the process, one must decide on how many models should be considered at a time. A number of strategies have been developed such as binary ladder integration, N-ary integration, balanced binary strategy, and mixed strategies. The strategy chosen will often be depending on the organisational setting for the modelling project.
2. Viewpoint comparison: Includes identifying correspondences and detecting conflicts among the viewpoints. Some types of conflict that may be detected are
 - Naming conflicts: Problems based on the use of synonyms and homonyms.
 - Type conflicts: That the same statements are represented by different concepts in different models.
 - Value conflicts: An attribute has different domains in two models.

- Constraint conflicts: Two models represent different constraints on the same phenomena.
3. Viewpoint conforming: Aims at solving the previously detected conflicts. Representations of statements in two different models can be classified as follows: Identical, equivalent, compatible, and inconsistent. To deal with such conflicts traditional approaches are mostly based on either transformational equivalence or they entrust the skill of the participants by providing only examples valid for the particular model.
 4. Merging and restructuring: The different models are merged into a joint model and then restructured. The latter involves checking the resulting model against criteria for empirical, semantic, and pragmatic quality.

5.7 Deontic Quality of a Conceptual Data Model

The remaining metrics from Moody belong to the level of deontic quality, in particular M-Flexibility, as the metrics are phrased.

- 13. Number of data model elements that are subject to change.
- 14. Probability adjusted cost of change
- 15. Strategic impact of change
- M-Integration (metric 27: Rating of ability to meet corporate needs)
- M-Implementability
 - 28. Development cost estimate
 - 29. Technical risk rating
- B-Pertinence relates to the data model being at the right level of detail (in particular the aspect of not being over-constrained)

Other possible relevant aspects taken from the discussion on the quality of an SRS in (Krogstie, 2001) are

- Annotated by relative importance
- Annotated by version
- Traceable
- Design-independent
- Unambiguous

6. CONCLUDING REMARKS

As with the quality of a software requirements specification (SRS) (Krogstie, 2001), we see some

benefit both for SEQUAL and for a framework for the quality of a conceptual data models by performing this kind of exercise:

- The eight areas of Moody can be argued to be more clearly conceptualised through this exercise, as exemplified with that metrics for the same area in the framework of Moody is positioned at different quality levels in SEQUAL.
- The overview of Moody has few points in the areas of physical, empirical, and social quality.
- The work of Batini has few points in the areas of physical, syntactic, pragmatic, and social quality.
- The work by Moody on the other hand enrich the areas of semantic and deontic quality for this type of models.

Thus, we can conclude that both research questions described in section 1 is answered positively. Future work will be to devise more concrete guidelines and evaluate the adaptation and use of these empirically in projects including conceptual data modelling. Some generic method-guidelines exist for the SEQUAL framework, which can be specialized for the quality of conceptual data models, but also keeping in mind that the particular context for a modelling project might result in that some areas are more important than others due to specific goals of modelling. Analytically it will be interesting to look more on the relationship between the quality of a conceptual data model and the quality of corresponding logical and physical data models, and also combine the work on data quality (data being looked upon as a model on the instance level) and quality of the corresponding data model. Another area is to look at the quality of combinations of data, process and rule-models.

REFERENCES

- Batini C., Scannapieco M. 2006. *Data Quality: Concepts, Methodologies and Techniques* Springer
- Battista G, Eades P, Tamassia R, Tollis IG 1994. Algorithms for drawing graphs: An annotated bibliography. Technical report, Brown University, June
- Booch G, Rumbaugh J, Jacobson I 2005. *The Unified Modeling Language: User Guide Second Edition*. Addison-Wesley
- Chen PP 1976. The entity-relationship model: Towards a unified view of data. *ACM Transactions on Database Systems*, 1(1):9–36, Marc
- Davies I, Green P, Rosemann M, Indulska M, Gallo S. 2006. How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering* 58 358–380
- Davis, A., Overmeyer, M., Jordan, S., Caruso, K., Dandashi, J., Dinh, F., Kincaid, A., Ledebor, G., Reynolds, G., Sitaram, P., Ta, P., and Theofanos, M.:1993 Identifying and Measuring Quality in a Software Requirements Specification. In: *Proceedings of the First International Software Metrics Symposium* pages 141-152
- Falkenberg ED, Hesse W, Lindgreen P, Nilsson BE, Oei JLH, Rolland C, Stamper RK, Assche FJMV, Verrijn-Stuart AA, Voss K 1996. A Framework of information system concepts - The FRISCO Report, IFIP WG 8.1 Task Group FRISCO
- Francalanci C, Pernici B 1993. View integration: A survey of current developments. Technical Report 93-053, Politecnico de Milano, Milan, Italy
- Gane, C. and Sarson, T.1979. *Structured Systems Analysis: Tools and Techniques*, Prentice Hall Professional Technical Reference
- Goodman N 1976. *Languages of Art: An Approach to a Theory of Symbols*. Hackett, Indianapolis
- Halpin T, Curland M 2006. Automated Verbalization for ORM 2. In : *Proceedings OTM*
- Hepp, M., Roman, D.2007. An Ontology Framework for Semantic Business Process Management. In: 8th International Conference Wirtschaftsinformatik, Karlsruhe, Germany, 2007, pp. 423-440 2
- Krogstie, J. 1999. Using Quality Function Deployment in Software Requirements Specification. Paper presented at the Fifth International Workshop on Requirements Engineering: Foundations for Software Quality (REFSQ'99), Heidelberg, Germany, June 14-15
- Krogstie, J. 2001. A Semiotic Approach to Quality in Requirements Specifications. Paper presented at the IFIP 8.1. Working Conference on Organizational Semiotics, Montreal, Canada, 23-25 July
- Krogstie, J. 2003. Evaluating UML Using a Generic Quality Framework. In: Favre L (ed) *UML and the Unified Process*. IRM Press, pp 1-22
- Krogstie, J. 2008. Integrated Goal, Data and Process modeling: From TEMPORA to Model-Generated Work-Places. In: Johannesson P, Söderström E (eds) *Information Systems Engineering From Data Analysis to Process Networks*. IGI, pp 43-65
- Krogstie, J. 2012. Model-based Development and Evolution of Information Systems: A Quality Approach
- Krogstie, J., Arnesen, S. 2004. Assessing Enterprise Modeling Languages using a Generic Quality Framework. In: Krogstie J, Siau K, Halpin T (eds) *Information Modeling Methods and Methodologies*. Idea Group Publishing
- Krogstie, J., Sindre, G. 1996. Utilizing deontic operators in information systems specifications. *Requirement Engineering Journal* 1:210-237
- Lillehagen, F., Krogstie, J. 2008. *Active Knowledge Modeling of Enterprises*, Springer-Verlag New York

- Moody, D. L., Shanks, G. G.: What Makes a Good Data Model? 1994. Evaluating the Quality of Entity Relationship Models. In: Proceedings of the 13th International Conference on the Entity-Relationship Approach (ER'94), pages 94-111, Manchester, England
- Moody DL. 1998. Metrics for Evaluating the Quality of Entity Relationship Models. In proceedings of the Seventeenth International Conference on Conceptual Modelling (ER '98), Singapore, Elsevier Lecture Notes in Computer Science, November 16-19
- Moody DL Shanks G. 2003. Improving the quality of data models: empirical validation of a quality management framework *Information Systems* 28 (6) September 2, Pages 619-650
- Moody DL 2005. Theoretical and practical issues in evaluating the quality of conceptual models: Current state and future directions. *Data and Knowledge Engineering* 55 243-276
- Nelson HJ, Poels G, Genero M, Piattini M 2011. A conceptual modeling quality framework. *Software Quality Journal*
- Nöth W 1990. *Handbook of Semiotics* Indiana University Press
- OMG 2011. BPMN v2 Specification. Technical report, OMG (<http://www.omg.org/>), <http://www.omg.org/spec/BPMN/2.0/> January
- Recker, J., Rosemann, M., Krogstie, J. 2007. Ontology-versus pattern-based evaluation of process modeling language: A comparison. *Communications of the Association for Information Systems* 20:774-799
- Sedera, W., Rosemann, M., Doebeli, G. 2005. A Process Modelling Success Model: Insights from a Case Study. In: 11th European Conference on Information Systems, Naples, Italy
- Tamassia R, Di Battista G, Batini, C 1988. Automatic graph drawing and readability of diagrams. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):61-79, January
- Tsichritzis, D, Klug, A 1978. The ANSI/X3/SPARC DBMS Framework. *Information Systems*, 1978. 3: 173-191
- User Requirements Notation (URN) 2012. <http://jucmnav.softwareengineering.ca/ucm/bin/view/UCM/DraftZ151Standard> Last accessed Oct. 21. 2012