Evaluating the Functionality of Conceptual Models

Kashif Mehmood[1,2] and Samira Si-Said Cherfi[1]

[1] CEDRIC-CNAM, 292 Rue Saint Martin, F-75141 Paris Cedex 03, France
[2] ESSEC, Avenue Bernard Hirsch B.P. 50105, 95021 Cergy-Pontoise Cedex, France
Kashif.Mehmood@essec.fr, samira.cherfi@cnam.fr

Abstract. Conceptual models serve as the blueprints of information systems and their quality plays decisive role in the success of the end system. It has been witnessed that majority of the IS change-requests results due to deficient functionalities in the information systems. Therefore, a good analysis and design method should ensure that conceptual models are functionally correct and complete, as they are the communicating mediator between the users and the development team. Conceptual model is said to be functionally complete if it represents all the relevant features of the application domain and covers all the specified requirements. Our approach evaluates the functional aspects on multiple levels of granularity in addition to providing the corrective actions or transformation for improvement. This approach has been empirically validated by practitioners through a survey.

Keywords: Conceptual Model Quality, Functional Quality, Quality Metrics, Quality Evaluation, Quality Improvement.

1 Introduction

An information system is designed to answer the user's requirements. Therefore, users expect it to deliver all the required functionalities, for which it has been designed, correctly. It has now been widely agreed that the quality of the end-system depends on the quality of the Conceptual Models (CM). These CMs are designed as part of the analysis phase and are the basis for further design and implementation. Thus, if there are errors and deficiencies in the CMs then they are propagated along the development process. These errors are more expensive to fix once the system is developed and deployed. For these reasons, different methodologies propose different methods and guidelines to ensure a certain degree of quality to the produced deliverables. These guidelines aim to make the developed models correct, consistent and complete with respect to the specified requirements. Some of these criteria are relatively simple and easy to check such as syntactic correctness since it relates to the used notation. Whereas, some characteristics are more difficult to verify and to ensure. For example, verifying that the designed CMs cover all the user requirements or verifying that the different designed models are consistent are more difficult. Indeed, a "good" conceptual model should respond to some characteristics that we

C.A. Heuser and G. Pernul (Eds.): ER 2009 Workshops, LNCS 5833, pp. 222–231, 2009.

refer to in this article as functionality. A conceptual model, with respect to functional quality, should: (i) cover all the requirements by proposing suitable functions, (ii) not propose functions out of the system scope, (iii) be consistent, (iv) reuse common functions if possible and (v) be reliable. However, the translation of the requirements into conceptual model depends heavily on the degree of expertise of the analyst. For this reason, we propose a quality approach to help the evaluation and improvement of the functional quality of conceptual models.

In this paper, we propose a quality model for functional quality. This model proposes a set of quality attributes for functional quality and a set of metrics to measure these attributes. Moreover, our quality evaluation is enriched with corrective actions provided to the designer, leading to a quality guided modeling process. The rest of the paper is organized as follows. Section 2 is a brief state-of-the-art. Section 3 describes our quality model for functional quality. A first validation based on a survey is described in Section 4. Section 5 concludes and mentions future research directions.

2 Literature Review

Research in software quality is rather mature and has produced several standards such ISO 9126 [1]. This standard defines a set of six characteristics to describe and to evaluate software quality. These characteristics are Functionality, Reliability, Usability, Efficiency, Maintainability and Portability. More precisely, Functionality is defined as a set of attributes that expect the existence of a set of functions satisfying the stated requirements.

In the domain of conceptual modeling, research on quality evaluation is rather young. The first structured approach dates back to the contribution of [2]. They were the pioneers in proposing quality criteria relevant to conceptual schema evaluation (completeness, correctness, minimality, expressiveness, readability, self-explanation, extensibility, normality). In [3], the quality of schemas is evaluated along three dimensions: syntax, semantics and pragmatics. Syntactic quality refers to the degree of correspondence between the conceptual schema and its representation. The semantic quality refers to the degree of correspondence between the conceptual schema and the real world. Finally, the pragmatic quality defines the degree of correspondence between a conceptual schema and its interpretation, which can be defined as the degree to which the schema can be understood. [4] identified the quality framework proposed in [3] to be the only one having a theoretical basis and an empirical foundation. In the context of Business Process Reengineering (BPR), the authors in [5] proposed a four dimensional framework for evaluating models and tools. Functionality is listed as the first dimension in their framework and is defined by the following criteria: expressiveness, structuring, formal/methodological support and relevance of concepts. The second dimension is ease of use. The third dimension is BPR trajectory and finally a general dimension related to tool price and customer support. [6] have reviewed existing frameworks on conceptual modeling quality and found lack of generalizability among the frameworks and lack of collaboration between researchers and practitioners.

The authors in [7] stepped ahead and reviewed cognitive mapping techniques to improve the quality of conceptual models.

To summarize, we could say that conceptual modeling is still considered as an art, which is poorly supported, by methods and tools. Our vision of functionality considers both the requirements and their coverage in the future system, and the modeling principles emerging from analysis and design practices. We argue that functionality of conceptual models cannot be evaluated on the same attributes as those used for software functionality evaluation as the software is not at hand yet. Moreover, as the conceptual modeling occurs early in the development process, it is important to make the emphasis on both the coverage of requirements and the fulfillment of good analysis and design principle.

This paper is a step forward in the description of conceptual models functionality. Moreover, we propose metrics to measure functionality and a set of transformation rules to improve it. These proposals are inserted in a generic quality model that can be applied not only to functionality but also to other criteria. The next section describes this quality model.

3 Quality Model for Functionality

Most of the end-users evaluate their Information systems (IS) based on its functionality. It has been noticed that majority of the IS change-requests results due to deficient functionalities in the information systems such as the lack of desired functionality within a system etc. Similarly, Studies show that defect detection in the early stages of the application development can be 33 times more cost effective than testing done at the end of development [8]. Therefore, it is very effective to catch the defects much earlier in the design phase. Conceptual models serves as the communicating mediator between the end user and the development team. Hence if the conceptual models are scanned for defects and the defects be corrected then it is likely to reduce the number of change requests for the end system. In this paper, we propose a feedback driven quality approach for the functional aspects of the conceptual models. We tend to detect and correct the functionality driven errors in the earlier stage of designing. For this, we have proposed a set of attributes that can evaluate the different aspects of conceptual models with respect to user's desired functionality. Our approach is unique in a way that it not only detects the errors but also provides corrective suggestions or transformations for their rectification.

3.1 Meta-model

In this article, we propose a meta-model driven solution that is generic and simple (Fig. 1). The meta-model starts by defining a quality goal that in this article is about improving the functional aspects of the conceptual models. Then respective quality dimensions are identified and similarly goal specific quality attributes and the corresponding metrics are identified. Once the metrics are calculated, corresponding transformations can be used for improving the quality of the conceptual model based on the quality goal.

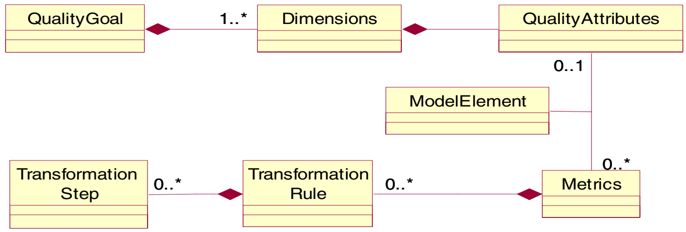

Fig. 1. The underlying meta-model

3.2 Defining a Quality Model for Functionality

In this article, we are interested in evaluating the quality of conceptual model with respect to its functional aspects. Thus, the quality goal is to "evaluate the functionality of the conceptual model". We can use the meta-model in figure-1 to instantiate our goal specific model for evaluation. One of the possible models for the problem in question can be that of Figure-2. The first level of the tree is the functionality dimension. It is composed of six attributes (second level). The third level proposes metrics to evaluate these attributes. The last level suggests transformation rules to improve these quality attributes. This hierarchy relates to the quality goal "Improve the functionality".

Quality Attributes for Functionality. Our solution suggests employing the functionality and its constituting attributes and metrics for evaluating the conceptual model based on the above mentioned quality goal of improving the functional aspects.

Functionality dimension consists of the set of attributes responsible for evaluating the model quality based on functional aspects. These attributes are, directly or indirectly related to the functional quality of the future product and addresses issue that could lead to functional changes in the final product. Furthermore, these attributes tries to identify the key problems that can hamper the functionality of the final product. Some of the attributes that can be used for our quality goal are:

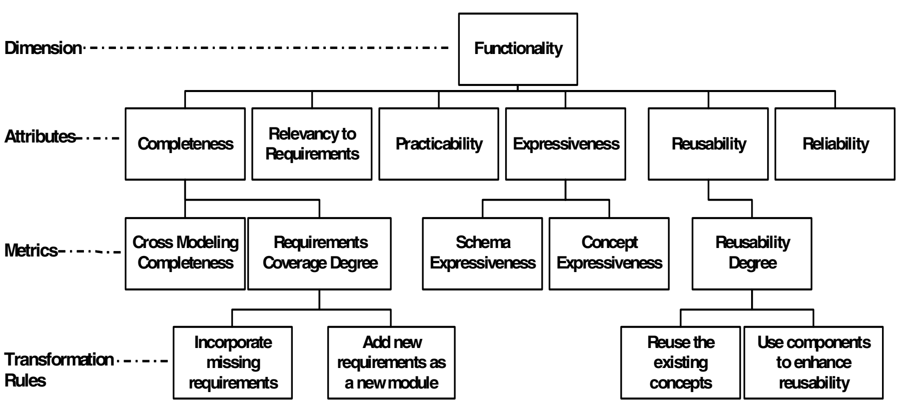

Fig. 2. A quality model for Functionality

Completeness. This attribute is based on the coverage of user requirements. It will try to evaluate the quality by comparing the conformance between concepts depicted in the conceptual model and the ones expressed by the users through the requirements. Furthermore, this attribute can be used to compare completeness among several schemas modeling the same reality. A schema can be considered complete if it covers all the modeling elements present in other schemas representing the same reality. This attribute can use collaboration patterns [9] to enhance the chances of model completeness. Moreover, this attribute can also evaluate whether the number of concepts present in the model corresponds to the number of concepts demanded by the user in their requirements.

Reusability. This attribute has been widely recognized and appreciated in the Object Oriented Paradigm. Reusability is considered a major opportunity for improving quality and productivity of systems development [3]. We choose this attribute to evaluate the quality of the model in twofold: First, to checks whether the model employs the previously developed models (e.g. use of existing modules) and secondly to check whether this model can be reused in future (for example to check if this model is specific or generic). Such an attribute will help in speeding up the process of modeling. Some studies suggest that reusability is feasible only if planned at the design stage because of loss of generalizability at subsequent stages [3]. Reusability is important in our model since it enhances the system's functional reliability because the reused component/module has been tested multiple times therefore errors and deficiencies would have been rectified during its maturity cycle.

Relevancy to requirements. This attribute is different from "Completeness" in a way that it is employed for finding the relevancy between the concepts present in the model and the ones required by the users. It will help in removing the irrelevant concepts present in the model thus will implicitly affect the complexity and functionality dimensions.

Practicability. This attribute is based on the notion of feasibility of the model. It verifies whether the model employs the concepts or elements that are realistic and can be materialized. For example, there can be some models that require unprocurable sophisticated technology for implementation.

Reliability. A system is reliable if it is not prone to failure. It is important to consider this attribute at the conceptual level as failure could be hardware or software failure. The software failures are generally caused by errors that could result from analysis decisions. Consequently, designers must design reliability in the system by; reusing reliable components, designing integrity constraint to ensure data integrity, facilitating it testability, etc.

Expressiveness. This attribute evaluates the expressiveness of a model. A model is expressive if it represents users' requirements in natural way and is understandable without additional explanation. This attribute evaluates whether the employed concepts are expressive enough to capture the main aspects of the reality. E.g. Inheritance link is more expressive than an association. So the more the expressive concepts are used, the more the schema will be expressive. Furthermore, this attribute evaluates the expressiveness by validating whether the existing notations are used to increase the expressiveness or not. For example, it can verify whether the multiplicities are defined in an ER diagram or not.

Quality Metrics Quantifying Functionality. The above mentioned quality model for functionality lists some of the metrics that can be used to quantify our quality attributes. Due to space constraints, we are listing only some of the metrics that are available for functionality. A more complete and a more formal description of metrics could be found in [12, 13].

Requirements Coverage Degree. This metric is based on notion of completeness of user requirements. It has been widely accepted that if the requirements errors are detected earlier in the designing phase then the cost of their rectification gets much lower. This metric calculate the ratio between the concepts covered by the modeling elements in the conceptual schema and the ones expressed by the users through the requirements.

Cross Modeling Completeness. This criterion is used to compare completeness among several schemas modeling the same reality. A schema is considered to be complete if it covers all the modeling elements present in the other schemas. Thus, this metric calculates the ratio between the number of concepts present in the model and the union of all the distinct concepts present in all the schemas representing the same reality.

Reusability Degree. This metric calculates the ratio between the reused concepts and the total concepts present in the model.

$$\text{Reusability Degree} = \text{Reused Concepts} / \text{Total Concepts}$$

Overall Model Reuse. This metric is adopted from Basili's[10] metric for overall system reuse. It calculates the aggregated reuse of the whole model by summing the reuse of every individual concept in the model. This metric uses the following formula for calculation:

$$\text{Reuse (Model)} = \sum \text{Reuse (Concept)}$$

Where *Reuse (Concept)* includes the count of all the ancestors of that concept and the concepts aggregated by that concept.

Coupling Between Concepts. This metric is adopted from [11]. It calculates the number of other concepts to which a concept is coupled. Low value for this metric signifies that the model is modular and promotes encapsulation.

Concept Expressiveness. It measures whether the used concepts are expressive enough to capture the main aspects of the reality. For example, Inheritance link is more expressive then association. So the more the expressive concepts are used, the more the schema will be expressive.

Schema expressiveness. It measures the expressiveness of the schema as a whole. A schema is said to be expressive when it represents users' requirements in natural way and can be easily understood without additional explanation. This metric assigns the weights of every concept and then takes the ratio between the calculated total value of the schema and the union of all the schemas describing the same reality.

Transformation Rules for Improvement. Corrective action or transformation rules are the main strength of our proposed solution. Once the quality metrics are calculated, corresponding corrective actions or transformations can be proposed to optimize the model. Due to space constraints, we are just defining two correction actions for the above mentioned two quality metrics.

Requirements coverage degree. If the metric shows that the model doesn't cover all the user requirements then the corrective actions can include the following:

1. Incorporate all the uncovered requirements.
2. If the incorporation of the missing requirements demands major modification to the model and if the model is modular then a new module can be used to address these upcoming requirements and can be interfaced with the existing model.
3. If the incorporation of the missing requirements completely changes the model then the whole model must be retested for conformance.

Reusability degree. If 'reusability degree' metric shows a very low value for reusability then some of the corrective actions could be:

1. Search the model to find the concepts for which equivalent concepts exists in the repository for reusability.
2. Decompose the model into multiple independent modules to facilitate the reusability.

4 Empirical Support

A web-based survey was used to empirically validate the quality model for functionality. The purpose of this survey was twofold:

i. To serve as a validation exercise in providing feedback from professionals including practitioners over the efficacy of our quality model.
ii. To study the general practices and views of the professionals over the quality of conceptual models. This includes the identification of attributes or factors important to professionals for evaluating the quality of conceptual models.

As mentioned above, a web-based survey was formulated to conduct this study. This was a closed survey and was accessible through a special link, provided to the invited participants only to avoid unintended participants. This was a comprehensive survey containing 42 general questions and our model specific questions. However, all the questioned were directly related to the quality of conceptual models. These questions include the two feedback questions where the participants were required to mention the quality attributes/factors that in their view are crucial to the quality of conceptual models. Moreover, they were also required to identify their practice for comparing two conceptual models representing the same reality or modeling the same problem. They were required to identify and mention up to seven attributes/properties that they think they will employ in choosing the best model with respect to their perception of quality.

The survey provides the dictionary and instant help about the definitions and details of all the terms and concepts that were present in the survey including the definitions of all the attributes in our model. Respondents were asked to classify each of our quality attributes into 'directly related to quality', 'indirectly related to quality', 'not related to quality' and 'I am not sure'.

4.1 Sample

In total 179 professionals (including IS managers, IS developers, Researchers etc.) were contacted to complete the survey. However, 57 professionals completed the survey that resulted in the response rate of 31.8%. Among the received 57 responses, three were discarded due to errors in the provided data or incomplete information. Average age of the respondents was approximately 30 years and average modeling experience was 4 years and 3 months.

Respondents belong to different organizations ranging from small organizations having less than 50 employees to as big as having more than 1000 employees. Moreover, respondents were required to select their occupation from a list of fifteen pre-defined occupations.

4.2 Data Analysis

The collected data shows that 85% of the respondents consider the imposition of quality approach on the conceptual models to directly influence the quality of the final product. However, it is interesting to note that 87% of the respondents have never used any method or approach to evaluate the quality of conceptual models. This shows that despite the appreciation of importance of implementing quality approach, professionals do not employ any methods to improve the quality. This behavior can be due to the gap between research and practice. To date there does not exist any quality framework for conceptual models that is standardized and comprehensive enough to accommodate the requirements of the practitioners. However, our proposed model is unique in a way that it is generic, simple and easy to implement. Moreover, the proposed approach follows a hierarchy of different quality levels starting from a quality goal and ending at the corrective suggestions.

As mentioned above, respondents were asked to provide feedback over the efficacy of the above mentioned attributes of our model. They were required to mark these attributes into either 'not related to quality', 'I am not sure', 'directly related to quality' or 'indirectly related to quality'. However since the last two options affirm that the attribute is related to quality therefore we have merged these two options as one to have a clear distinction between the attributes that are related to and not related to quality. The responses are summarized in the Table 1. All the values are in percentages of the responses and are rounded off to the nearest tenth digit. Table-1 should be read as, for example, 75.9 % of the respondents think that 'completeness' is related to quality against 7.4% that think 'completeness' is not related to quality. Similarly, 13% of the respondents declare their inability to categorize 'completeness' in any of four classes.

Table 1. Respondents' feedback on the quality model for Functionality

Attributes	NOT Related to Quality	Related to Quality	Not answered	I am not sure
Completeness	7.4	75.9	3.7	13
Reusability	16.7	64.8	3.7	14.8
Relevancy	3.7	83.3	3.7	9.3
Reliability	11.1	79.6	3.7	5.6
Practicability	9.3	77.8	3.7	9.3
Expressiveness	0	74.1	3.7	22.2

After viewing the above feedback, we can say that the attributes in the functionality dimension are well identified and represent the attributes and factors required by professionals. However, respondents have also identified some attributes that they think are important to quality such as Validity and Degree of abstraction. These attributes will be incorporated in our approach after validation.

5 Conclusion and Implications for Further Research

The functional aspects of conceptual models cannot be evaluated on the same attributes as the ones used for software functionality as the software is not at hand yet. We propose to optimize the overall quality of the IS by ensuring the functional aspects of the conceptual models during the analysis and design phase. In this paper, we have addressed the concept of functionality at the conceptual level. Our approach emphasizes the coverage of requirements and the fulfillment of good analysis and design principles. Our main contribution is a model for evaluating and improving the functionality of conceptual schema. This model has been instantiated from our meta-model and defines a set of quality attributes for functionality refinement. The strength of our approach lies in the post evaluation feedback in the form of corrective actions or transformations. The functionality model has been empirically validated by the professionals through a web based survey. The empirical results show that the respondents consider the identified attributes to be related to functional quality. Our meta-model can also be used to evaluate the conceptual models on other user specified quality goals.

Future directions of this work include:

- The extension and enrichment of the current quality model;
- The development of an environment implementing the proposed quality approach;

References

1. Software Engineering - Product quality - Part 1: Quality model 2001(ISO/IEC 9126)
2. Batini, C., Ceri, S., Navathe, C.: Conceptual Database Design: An Entity-Relationship approach, p. 496. Benjamin/Cummings Publishing Company Inc. (1992)
3. Lindland, O.I., Sindre, G., Sølvberg, A.: Understanding Quality in Conceptual Modeling. IEEE Software 11(2), 42–49 (1994)

4. Maes, A., Poels, G.: Evaluating Quality of Conceptual Modeling Scripts Based on User Perceptions. Data & Knowledge Engineering 63(3), 701–724 (2007)
5. Teeuw, B., Van Den Berg, H.: On the Quality of Conceptual Models. In: Embley, D.W. (ed.) ER 1997. LNCS, vol. 1331. Springer, Heidelberg (1997)
6. Moody, D.L.: Theoretical and Practical Issues in Evaluating the Quality of Conceptual Models: Current State and Future Directions. Data & Knowledge Engineering 55, 243–276 (2005)
7. Siau, K., Tan, X.: Improving the quality of conceptual modeling using cognitive mapping techniques. Data & Knowledge Engineering 55(3), 343–365 (2005)
8. Walrad, C., Moss, E.: Measurement: The Key to Application Development Quality. IBM Systems Journal 32(3), 445–460 (1993)
9. Ribbert, M., Niehaves, B., Dreiling, A., Holten, R.: An Epistemological Foundation of Conceptual Modeling. In: The Proceedings of the 12th European Conference on Information Systems, Turku, Finland (2004)
10. Basili, V.R., Rombach, H.D., Bailey, J., Delis, A.: Ada Reusability and Measurement. Technical Report, Institute for Advanced Computer Studies and Department of Computer Science at the University of Maryland College Park (1990)
11. Chidamber, S.R., Kemerer, C.F.: A Metrics Suite for Object Oriented Design. IEEE Transactions on Software Engineering 20(6), 476–493 (1994)
12. Cherfi, S.S., Akoka, J., Comyn-Wattiau, I.: Measuring UML Conceptual Modeling Quality-Method and Implementation. In: Pucheral, P. (ed.) The Proceedings of the BDA Conference, Collection INT, France (2002)
13. Cherfi, S.S., Akoka, J., Comyn-Wattiau, I.: A Framework for Conceptual Modeling Quality Evaluation. In: The Proceedings of ICSQ 2003 (2003)