

A MetaModel for Quality Software Based on the MDA Approach

Youness BOUKOUCHI¹, Adil KHAMAL², Abdalaziz MARZAK³, Hicham MOUTACHAOUIK⁴

¹²³ Ben M'Sik Faculty of science, University Hassan II, PB 7955 Casablanca, Morocco

⁴National School of Arts and Trades, PB 7955, University Hassan II, Casablanca, Morocco

Abstract—Today, technology and IT platforms are in continuous exponential growth, however, this growth has created several problems related to portability, reusability, interoperability but most important of all is the fact that it is related to the software quality. From these facts, the OMG proposed the Model Driven Architecture (MDA), which is based on the models as independent primary objects of any platform, and which are transformed thereafter into other objects specific to each platform (code, dB count, XMLFile,...). Therefore, to assess and ensure the quality of software, stakeholders (designer, developers, users , ...) in this domain, need a quality model , the most difficult thing to do is to choose it from a diversity of models taking into account the rarity of the meta-models. Based on the MDA approach, this article will introduce a software quality and quality models, a comparison of model structures and finally, the proposal of our meta-model of software quality.

Keywords: MDA, Model Quality, Metamodel, Metrics, Software Quality.

1. INTRODUCTION

Today, technology and IT platforms are in continuous exponential growth , however, this growth has created several problems related to portability, reusability, interoperability but most important of all is the fact that it is related to the software quality. The models of quality suggested are general in the description of software quality in particular when we want to technically define and describe the metrics.

So the American association Object Management Group (OMG) whose objective is to standardize and promote the model object in all its forms, proposed the Model Driven Architecture (MDA), which is based on the models as independent primary objects of any platform which are transformed thereafter into other objects specific to each platform (code, dB count, XMLFile,...).

Therefore, to assess and ensure the quality of software, stakeholders (designer, developers, users, ...) in this domain, need a quality model , the most difficult thing to do is to choose it from a diversity of models taking into account the rarity of the meta-models. Based on the MDA approach, this article will introduce the software quality and quality models, a comparison of model structures, and finally the proposal of our meta-model of software quality.

This paper is organized as follows: section 2 presents the concepts of MDA and software quality, Section 3 presents a comparison of the structures of software quality models, Section 4 presents our meta-model quality, section 5 presents conclusions and describes future work.

2. SOFTWARE QUALITY AND MODELS

1. Software Quality

Software quality is the most important element in the development of software, because the quality could reduce the cost of maintenance, software testing, etc... Quality has very different meanings for customers, users, managers, developers, testers, etc... Many institutes and organizations have their own definitions of software quality and also quality models. Below some definition of software quality [2]:

- ISO 9126 : Is a set of attributes of a software product which describes and evaluates the quality.
- ANSI : Quality is the totality of features and characteristics of a product or service that relies on its ability to meet the specific needs.
- IEEE (IEEEStd 729-1983) : The totality of features and characteristics of a software product that influence on its ability to meet specific needs.

In the most general sense, software quality can be defined as: An effective process for software development, applied in a manner that creates a useful product and delivers measurable value for those who produce it and those who use it.

2. Models And Metamodels

The model-driven architecture (MDA) [14] is an approach to software development which emphasizes the use of models in the specification, development, analysis, verification and management systems. MDA places models at the center of software engineering process, it is a form of generative engineering, in which all or a part of software is generated from models (Figure 1).

Each model usually responds to a problem, independently from the rest of the technical issues involved in building the system. A model is written in the language of their meta-model. A meta-model describes the concepts of language, the relationship between them, the mapping rules and transformation of model elements to comply with the rules of the domain.

As the number of quality models and their importance grow, the need to measure and evaluate the quality model by meta-models is becoming increasingly relevant. However, there is a lack of meta-models repositories that allow generation of these models. International standards (in particular ISO and IEEE) have to state to be too general

to treat the specific characteristics of the meta-models because of their double natures: they are models and at the same time they are languages of modeling.

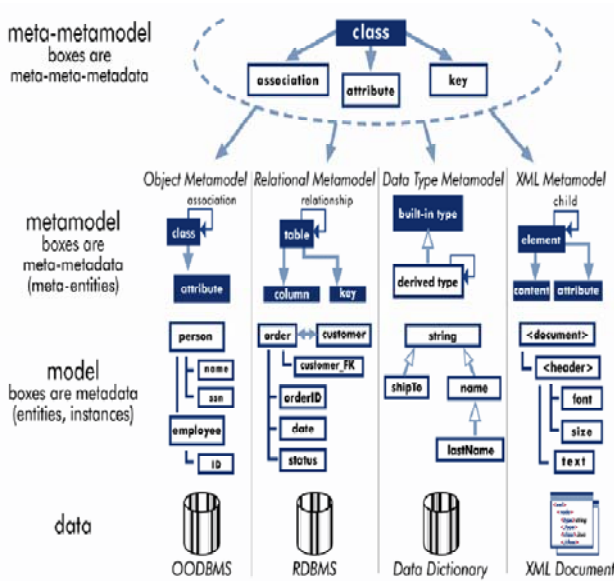


Figure 1: Process model development by OMG

3. STRUCTURE OF QUALITY MODELS

1. Quality Models

ISO/IEC 9126-1 defines a quality model as a "framework which explains the relationship between different approaches to quality". Quality models decompose in hierarchical elements. An approach to quality is to decompose quality in Factors, Sub-factors, and criteria. Evaluation of a program begins with measuring each quality criteria with numerical value from metrics. Then, each quality sub-factors is assessed using their criteria. Finally, numerical values are assigned to quality characteristics from their quality sub-factors [2]. Below are some models and methods of software quality [1]:

a) MCCALL MODEL

McCall model presents eleven criteria grouped into three visions: operations, revisions and product transitions. This model is allocated as follows: factors, criteria and metrics.

b) BOEHM MODEL

Boehm's model is similar to the model of McCall. It also presents a hierarchical quality model structured around four levels: high-level characteristics, intermediate-level characteristics, primitive-level characteristics and metrics.

c) DROMEY MODEL

Dromey model is structured around a process focused on the relationship between quality attributes and sub-attributes, and the attempt to connect the properties of products with attributes. To create this new model, the main idea was to obtain a large model to satisfy various systems.

The levels of this model are defined as follows: the properties of the product, quality attributes, sub-attributes and metrics.

d) ISO 9126 ET 25000 MODEL

The model ISO9126 and ISO 25000, defines and describes a series of characteristic qualities of a software product (internal and external characteristics, characteristics of use) that can be used to specify the functional and non-functional requirements of customers and users.

Each characteristic is decomposed into sub-characteristics, and for each of them, the standard provides a set of metrics to put in place to assess the conformity of the product developed from the requirements contained in the guidelines.

e) GQM APPROACH

GQM (Goal, Question, and Metric) [15] is an approach to software metrics that has been promoted by Victor BASILI, GQM defines a measurement model on three levels:

- Conceptual level (goal): A goal is defined for an object, for a variety of reasons, with respect to various models of quality, from various points of view and relative to a particular environment.
- Operational level (question): A set of questions is used to define models of the object of study and then focuses on that object to characterize the assessment or achievement of a specific goal.
- Quantitative level (metric): A set of metrics, based on the models, is associated with every question in order to answer it in a measurable way.

f) IEEE 1061-1998 APPROACH

This standard provides a methodology to establish quality standards implementation and validation process for measuring product quality. This method applies to all phases of software for the lifecycle of the software [3].

The software quality metrics framework (Figure 2) is designed to be flexible. It permits additions, deletions, and modifications of quality factors, quality sub-factors, and metrics. Each level may be expanded to several sublevels. The framework can thus be applied to all systems and can be adapted as appropriate without changing the basic concept.

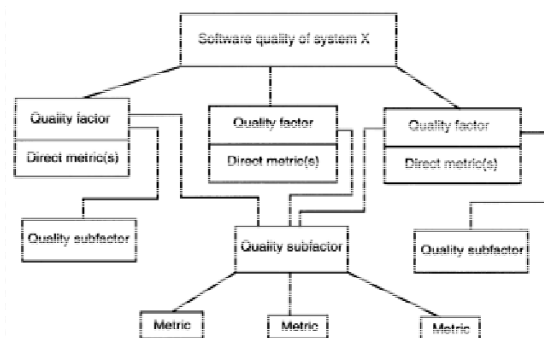


Figure 2 : Software quality metrics framework

TABLE 1: A COMPARISON BETWEEN THE STRUCTURES OF MODEL QUALITY SOFTWARE.

	Mac Call	Bohem	ISO	GQM	IEEE 1061	Dromey
Level 1	View	View	View	View	View	View
Level 2	Factors	high-level characteristics	Characteristics	goals	Factor	Product properties
Level 3	criteria	Intermediate level characteristics	Sub-characteristics	questions	Subfactor	Quality attributes
Level 4	-	primitive characteristics	Quality Attributes	-	-	sub-attributes
Level 5	Metric	Metric	Metric	Metric	Metric	Metric

2. A Comparison Between The Structure Of Models Quality

After analyzing the structures of these quality models, we can compare their structures (Table 1), and draw the following conclusions:

- All quality models (MacCall, ISO, Bohem), and approaches of definition of software quality (IEEE, GQM, Dromy) define their models according to a specific point of view (user, developer, designers, etc..).
- All views are decomposed into major quality characteristics called: Factor, Objectives, etc..
- These major characteristics are decomposed into sub-characteristics (called criterion or question in some model). these sub-characteristics may also contain sub-characteristics called in some models, attributes, basic or primitive characteristics ;
- And finally in each structure of model, there are the metrics, they are the basic elements presented by a quantitative value.

4. A METAMODEL FOR SOFTWARE QUALITY

1. Metamodel Repository

From the comparison above between quality models (Table.1), we proposed a meta-model repository (Figure 3) all existing models of quality, it can generate models as ISO9126, MacCall,... or generate personal models according to the requirements of the designer (User, developer, etc.).

This generation based on our meta-model used to generate a model in the form of a file XML, which can be a point of communication between all stakeholders in the quality system, and without problems of technical specifications of the platforms, XML technology is the basis for all platforms.

2. Metamodel Structure

This meta-quality model (Figure 3) is divided into hierarchical elements. It structure quality is divided into three levels: view, characteristic and metric, whose characteristics can be divided into several sub-characteristics and so on.

- a) Overview (Point of view): Quality can be perceived with various points of view, differences

of views are mainly due to the fact that the project has many stakeholders, each stakeholder perceives the quality of its manner, what implies a prospect focused on the specific requirements of stakeholder towards the system.

- b) Characteristic: After the view, we find the characteristics, (called Factors, Goals, Properties, etc), these characteristics are broken up into several under-characteristics until arrived in granular indecomposable characteristics and which are directly measurable by metrics.
- c) Metric: A metric used to measure and evaluate a characteristic by values.

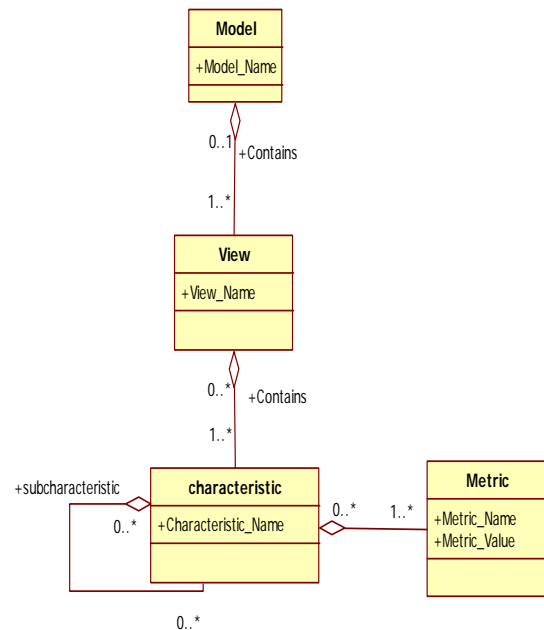


Figure 3: Meta-model for software quality

3. Mapping Rules

The mapping of our meta-model is presented in Table 2 which explains the passage of the elements of our meta-model to other quality models. This mapping shows the power of our meta-model, respecting the specificity of each quality model (ISO, MacCall, IEEE, ...) and the possibility of generating other personal model.

TABLE 2: MAPPING RULES OF META-MODEL

Metamodel		Mac Call	Bohem	ISO	GQM	IEEE 1061	Dromey	personel model
View		View	View	View	View	View	View	View
characteristics	characteristics	Factors	High characteristics	Characteristics	goals	Factors	Product properties	characteristics
	Sub-characteristics	critérias	Intermediate characteristics	Sub-characteristics	questions	Subfactor	Quality attributes	Sub-characteristics
	Sub-subcharacteristics	-	primitive characteristics	Quality Attributes	-	-	sub-attributes	Sub-subcharacteristics
	N sub subcharacteristics	-	-	-	-	-	-	N sub-subcharacteristics
Metric		Metric	Metric	Metric	Metric	Metric	Metric	Metric

1. CONCLUSION

The proposed meta-models will help stakeholders describe the quality of their models as well as instantiate and communicate them to any platform without taking into account technical specifications. This work is considered the first step in the development of metamodel quality system, we still have to develop the library and syntax of metric.

REFERENCES

[1] Comparative Study of Software Quality Models, Youness BOUKOUCHI et al, the International Journal of Computer Science Issues, April 2013.
 [2] A Quality Model for Design Patterns .Khashayar Khosravi and Yann-Gael Gueheneuc, Summer 2004
 [3] IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications
 [4] IEEE Std 1061™-1998 (R2009) IEEE Standard for a Software Quality Metrics Methodology
 [5] IEEE Std 982.1™-2005 (Revision of IEEE Std 982.1-1988) IEEE Standard Dictionary of Measures of the Software Aspects of Dependability

[6] ISO/IEC TR 9126, Software engineering –Product quality – Part 1,2,3. 2002-03-15.
 [7] “Enterprise Information Integration and the OMG’s MDA and MOF”, Randall M. Hauch, OMG's Third Workshop on UML for Enterprise Applications: Model Driven Solutions for the Enterprise October 21-24, 2002
 [8] “A Metamodel for Specifying Quality Models in Model-Driven Engineering”, Parastoo Mohagheghi and VegardDehlen, Engineering Research Institute, University of Iceland , 2008
 [9] Definitions and Approaches to Model Quality in Model-Based Software Development – A Review of Literature, PARASTOO MOHAGHEGHI, VEGARD DEHLEN, TOR NEPLE, Engineering Research Institute, University of Iceland , 2009
 [10] 12 Steps to Useful Software Metrics, Linda Westfall,The Westfall Team, westfall@idt.net
 [11] La mesure des modèles par les modèles : une approche generative, Martin Monperrus, octobre 2008.
 [12] Software Engineering Metrics: What Do They Measure and How Do We Know? Cem Kaner and Walter P. Bond, 10TH INTERNATIONAL SOFTWARE METRICS SYMPOSIUM, METRICS 2004.
 [13] Modèles de mesure de la qualité des logiciels, Karine Mordal, Jannik Laval et Stéphane Ducasse, November 7, 2011
 [14] Understanding The Model Driven Architecture (MDA), Sinan Si Alhir, Fall 2003, Volume 11 N°3 Page 18, ISSN 1023-4918
 [15] THE GOAL QUESTION METRIC APPROACH, Victor R. Basili1 et al, Encyclopedia of Software Engineering, 2 Volume Set, 1994