# QuEF
## *An Environment for Quality Evaluation on Model-driven Web Engineering Approaches*

F. J. Domínguez-Mayo, M. Mejías, M. J. Escalona and A. H. Torres
*University of Seville, Seville, Spain*

Keywords: Model-Driven Engineering, Quality, Software Metrics, Model-Driven Web Engineering, Methodologies.

Abstract: Due to the high number and wide variety of methodologies which currently exist in the field of Model-Driven Web Engineering (MDWE) methodologies, it has become necessary to evaluate the quality of the existing methodologies to provide helpful information for the developers. Since proposals are constantly appearing, the need may arise not only to evaluate quality but also to find out how it can be improved. This article presents work being carried out in this field and describes tasks to define QuEF (Quality Evaluation Framework), which is an environment to evaluate, under objective measures, the quality of Model-Driven Web Engineering methodologies.

## 1 INTRODUCTION

Model-Driven Engineering (MDE) is a paradigm of software development that consists of the creation of models closer to a particular domain rather than concepts or a specific syntax. The domain environment specific to MDE for web engineering is called Model-Driven Web Engineering (MDWE). The Object Management Group (OMG) has developed the standard Model-Driven Architecture (MDA) which defines an architecture platform for proposals based on the Model-Driven paradigm[1].

The concept of platform independence appears frequently in MDA. Models may have the quality of being independent from the characteristics of any technological platform. By applying this paradigm, the lifecycle of a software system is completely covered, from requirements capture to its own maintenance, through the generation of the code. In recent years, the growing interest in the internet has led to the generation of a high number of proposals (W. Schwinger et al., 2008) which offer a frame of reference for the Web environment. On the other hand, there are a high number of approaches without standard consensus, a lack in the use of standards, and scarcity of both practical experience and tool support. In the face of this situation, an important need to assess the quality of existing methodologies arises. In this paper, therefore, an environment for the quality evaluation of Model-Driven Web methodologies based on MDA is proposed.

The paper is organized into the following sections. In Section 2 a global analysis of the situation is presented. Section 3 presents the problem, motivation and goal, and is intended to lay the basis of a framework that allows evaluate the quality of different methodological proposals. In Section 4 concepts such as framework and MDWE methodology are explained and the elements which define the Quality Evaluation Framework (QuEF) are provided. In Section 5, an example of applying the framework proposed with the NDT methodology is performed. Finally, in Section 6, a set of conclusions, contributions and possible future work are given.

## 2 RELATED WORK

A great amount of work has been published in the area of MDWE, as have numerous studies related with MDWE approaches (Escalona and Aragón, 2008), (Kroiβ and Koch, 2008). Along this line (Schwinger et al., 2008) must be considered where a well-defined fine-grained catalogue of more than 30 evaluation criteria about approaches is defined. Some work related with quality evaluation and

[1] http://www.omg.org



J. Domínguez-Mayo F., Mejías M., J. Escalona M. and H. Torres A. (2010).
QuEF - An Environment for Quality Evaluation on Model-driven Web Engineering Approaches.
In *Proceedings of the 5th International Conference on Software and Data Technologies*, pages 137-142
DOI: 10.5220/0002928501370142

software metrics are considered in this section. The idea of developing a MDE framework for evaluating quality has been applied in various studies of (Mohagheghi et al.), where it is stated that the quality of models is affected by the quality of different factors. From the methodological perspective, software measurement is supported by a wide variety of proposals, the ISO 15539 and IEEE 1061-1998 standards deserving special attention. As far as web metrics quality is concerned, in (Calero et al., 2005) some important metrics proposed for web information systems are classified.

# 3 PROBLEMS, MOTIVATION AND GOALS

The main goal of this research is to lay the basis of a QuEF that facilitates the quality assessment of different methodological proposals under some specific criteria. Today's modern web information systems are called to manage a huge amount of information and are difficult to develop and maintain. Hence, there is a need for the suitable design of MDWE methodologies and effective tools. To this end, our work concentrates on evaluating and comparing existing proposals. One aspect that must be considered is the use of an MDWE methodology and its influence on the final product quality. Nowadays, in the software industry, there is an even greater need to produce faster, cheaper software of higher quality.

# 4 DEFINING A QUALITY EVALUATION FRAMEWORK FOR MDWE METHODOLOGIES

In this work, an approach, or methodology, is a Model-Driven proposal for the development of web applications. It may provide a set of guidelines, techniques, processes and/or tools for the structuring of specifications, which are expressed as models. The QuEF is a basic conceptual structure composed of a set of elements used to evaluate MDWE methodologies. Therefore, an environment with a set of elements based on existing literature is proposed, where four components for the evaluation of the quality of MDWE methodologies can be seen:

- *Approach Characteristics Template:* This component would have the responsibility of describing the input methodology characteristics to be evaluated.
- *Thesaurus & Glossary:* This component would be responsible for improving the standardization of the access channel and communication between users of different MDWE methodologies.
- *Quality Model:* This component is responsible for providing the basis for specifying quality requirements with the purpose of evaluating quality.
- *Quality Evaluation Process:* This component would have the responsibility of carrying out the quality evaluation process.

## 4.1 QuEF for MDWE Methodologies

We present the steps for defining QuEF for MDWE methodologies. The main component for QuEF is the Quality Model. Concepts, tasks to be performed for each step, framework structuring and components which result for each step are described. In this work, a Quality Model is a set of characteristics, subcharacteristics and metrics, quality factors, quality attributes and the relationships between them, which provides the basis for specifying quality requirements and evaluating quality. In Figure 1, the Quality Model metamodel with the relations between the different elements in the Quality model are shown, and the elements are described and explained.

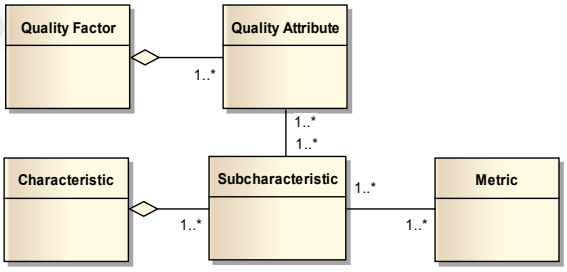


Figure 1: Quality Model Metamodel.

- *Quality Factor:* This is a higher-level feature that affects an item's quality. For example, a quality factor could be Usability, Functionality or Portability. Each quality factor and attribute in ISO 9126 is described in relation with a software product but in our particular case all quality factors and attributes are described in relation with approach characteristics.
- *Quality Attribute:* A quality attribute is a feature or characteristic that affects an item's quality (Syn: quality factor). In a hierarchy of quality

attributes, higher-level attributes may be called quality factors, lower-level attributes called quality attributes (IEEE 610). For example, Usability is defined for various quality attributes such as Learnability, Understandability, Operability, etc.

- *Characteristic*: This is a higher-level concept of an approach. It may be, for example, the software development process, models, metamodels, languages, tools, transformations or the quality assurance techniques.
- *Subcharacteristic*: This is a lower-level concept of an approach. For example, the Model-Driven Engineering characteristic may have various subcharacteristics such as, the Language Definition, Transformations and Trace Generation.
- *Metric*: In the Quality Model, metrics should indicate which quality attribute is affected by subcharacteristics and also the degree to which it is affected. For example, the evaluation may be via measuring quantitatively by metrics or subjective evaluation, inspections using checklists or interviewing the users.

Therefore, a *Quality Model* contains a minimal amount of *characteristics* and *subcharacteristics* through which any kind of MDWE approach can be evaluated. In order to define a *Quality Model*, it contains *association links* between the *subcharacteristics* and the *quality attributes*. These *association links* represent the dependencies between *subcharacteristics* and *quality attributes*. They show quality attributes which are affected by subcharacteristics or the areas of the methodology that will be significantly affected if the approach is changed. Association links may be based on proven and real-world experience. The impact of each subcharacteristic on quality attributes must be demonstrated and the requirements determined by real case study applications to a number of real projects. Hence, a quality factor has various quality attributes and a characteristic has various subcharacteristics, as is shown in Figure 1.

### 4.1.1 Identifying Quality Factors

A set of quality factors based on current literature, such as ISO/IEC 9126, IEEE and other standards which are adapted to MDWE methodologies, has to be identified, classified and hierarchical. The Quality Factors of an approach include Usability, Functionality, Reliability, Maintainability and Portability. Each quality factor and attribute in ISO 9126 is described in relation with a software product whereas in our study all quality factors and attributes would be described in relation with approach characteristics. In this work, *Reliability* is taken as an example of the quality factor. In ISO 9126, *Reliability* is a quality factor which is defined as: *The ability of the software product to perform its required functions under stated conditions for a specified period of time, or for a specified number of operations.* This definition could be adapted to more closely fit our specific domain: "*The ability of a characteristic approach to perform its required functions under stated conditions for a specified period of time, or for a specified number of operations*" or in a general way could be described as: "*A set of attributes that bear on the capability of a characteristic approach to maintain its level of performance under stated conditions for a stated period of time*".

### 4.1.2 Identifying Quality Attributes for each Quality Factor

For each quality factor, a set of quality attributes have to be identified. For example, quality attributes related with *Reliability* are described in the same way by adapting other definitions from ISO, IEEE, other standards and work already published. Some of these quality attributes may be described as:

- *Maturity*: The capability of a characteristic approach to avoid failure as a result of defects in the approach.
- *Recoverability*: The capability of a characteristic approach to re-establish a specified level of performance and recover a stated development point affected in case of failure.
- *Fault Tolerance*: The capability of a characteristic approach to maintain a specified level of performance in cases of faults (defects) or of infringement of requirements.
- *Compactness*: The state of being compact. A characteristic approach is more compact if it is getting closer to a specific domain.

### 4.1.3 Identifying Characteristics

In MDWE, models are refined progressively and transformed into new models or code with tools. Moreover, each methodology may define its development process and/or techniques. The idea is to characterize the whole MDWE process. The quality of methodologies in turn depends on the following Characteristics: The *Model-Driven Engineering*, the *knowledge of MDWE methodology users*, the *web modelling,* the *customization*

*modelling*, the *maturity* of a methodology, the *tool support* and the *quality assurance techniques.* Methodology users and developers use the available modelling languages, tools and processes and develop models based on their knowledge of the problem and their experience.

### 4.1.4 Identifying Subcharacteristics and Metrics for each Characteristic

For each subcharacteristic, a specification of its evaluation is necessary. For example, the evaluation may be carried out via quantitative measurement using either metrics or subjective evaluation, whereby checklists are used in inspections, or users and designers are interviewed, respectively. Metrics defined so far are qualitative metrics which indicate if the subcharacteristic is *Supported (S), Partly Supported (PS)* or *Not Supported (NS).*

### 4.1.5 Specifying Association Links between the Subcharacteristics and the Quality Factors

In this step, the association links between subcharacteristics and quality attributes have to be defined. A set of hypotheses has to be proposed to indicate which quality attribute is affected by each subcharacteristic. For example, Reliability is described as a set of quality attributes. These quality attributes could be affected by one of various subcharacteristics as shown in Table 1.

Table 1: Association links between realiabilty quality attributes and Tool support subcharacteristics.

| | | Reliability quality factor | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | Maturity | Recoverability | Fault Tolerance | Availability | Currency | Compactness | Relevancy |
| Tool Support characteristic | Creation Tool | X | | | X | X | X | X |
| | Analysis Tool | X | X | X | X | X | X | X |
| | Transformation Tool | X | | | X | X | X | X |
| | Composition Tool | X | | | X | X | X | X |
| | Test Tool | X | X | X | X | X | X | X |
| | Trace Tool | X | X | X | X | X | X | X |
| | Repository Tool | X | X | X | X | X | X | X |

Subcharacteristics and the relations between quality attributes are described below. The employment of Tool Support could give the approaches a broader application base. Our initial hypothesis is that it could bear influence on:

- *Maturity*: The use of tools can prevent faults in the use of a methodology because tthey make this methodology easier to handle.
- *Recoverability*: With subcharacteristics, certain mistakes can be detected and a stated point in the development can be recovered in case of faults.
- *Fault Tolerance*: Analysis, Test and Trace tool support help to make the use of a methodology more reliable in the case of faults.
- *Compactness*: Every subcharacteristic is getting closer to the MDWE specific domain.

The Quality Model component would then be refined and improved based on results, experience or current literature. Other subcharacteristics have to be proposed and they have to be associated with quality attributes. In this work, a set of Tool Support subcharacteristics and a set of hypotheses for linking these subcharacteristics to quality attributes of Reliability are proposed as an example. A consequence of having this matrix could be that in the case of improving a web development methodology, priority can be given to subcharacteristics which improve a greater number of quality attributes, and therefore, the quality process improvement is optimized.

### 4.1.6 Defining the Quality Evaluation Process

The *Quality Evaluation Process* component compares the information from each input *Approach Characteristics Template* with information from the *Quality Model*. The idea is to determine which aspect needs to be improved in MDWE methodology. The results provide an assessment report of the methodology and this may be used for comparison with the evaluation of other MDWE methodologies. Along these lines, Microsoft Excel is currently used for implementing a brief prototype for proofs. A spreadsheet is used to simulate the *Approach Template Characteristic* component where the user selects *subcharacteristics* according to a set of *metrics*. For example, the *metric* value is *1* if it is *supported*, *1/2* or the arithmetic mean of supported elements from among the total elements if it is *partly supported,* and 0 if it is not supported. The total value for the quality attribute is the number of values divided by the total *metrics* in the *subcharacteristic*. For every *quality attribute*, a quantitative value for each *subcharacteristic* is then calculated. A spreadsheet with a matrix which represents the *association links* between

*subcharacteristic* and *quality attributes* indicates if the *quality attribute* is influenced by the *subcharacteristic*. Finally, on another spreadsheet, graphics and total values of evaluation are shown.

# 5 EXAMPLE APPLICATION: NDT METHODOLOGY EVALUATION

NDT (Navigational Development Techniques) is a methodological approach oriented towards Web Engineering. It is an approach defined in the Model-Driven paradigm and it offers a suitable and easy-to-use methodological environment. With the use of NDT-Suite, NDT offers tool support for each phase of the complete life cycle of a software project. In the following evaluation of NDT, the extended revision supported by NDT-Suite is considered.

## 5.1 Applying the Approach Characteristic Template in NDT Methodology for Tool Support

The Approach Characteristics Template component has been applied using an implementation in Microsoft Excel. However, the Approach Characteristics Template component has not yet been fully developed, and only Tool Support characteristic, MDE characteristic and Maturity characteristic can be considered. In this example, only some metrics of Transformation Tool Support are shown in Table 2 as an example of the Template. Other subcharacteristics, such as Creation, Composition, Test, Trace, Repository Tool Support, are not shown but they have been considered in the evaluation process of this example in Section 5.2. An MDWE tool is a tool used to develop, interpret, compare, align, measure, verify and transform models or metamodels in the context of MDWE.

For example, with respect to the Transformation Tool Support subcharacteristic in the template, the synchronization tool metric PSM2PIM, PSM2CIM, Code2CIM and Code2PSM is Not Supported for NDT methodology as is shown in the Table 2. Other metrics are supported. Total metric results of Tool Support characteristics are shown in Figure 2. In the figure, black bars represent NDT result metric values for each subcharacteristic of the Tool Support characteristic.

Table 2: Metrics of Transformation Tool Support subcharacteristic.

| **TRANSFORMATION TOOL SUPPORT SUBCHARACTERISTICS AND METRICS** | | | |
|---|---|---|---|
| **Model-Driven Reverse Engineering or Synchronization Tool** | | | |
| This supports a Reverse Engineering Tool: A tool intended to transform particular legacy or information artifact portfolios into fully-fledged models. *S* | | | |
| This supports a synchronization tool between transformations such as: | | | |
| PIM2CIM | *S* | Code2CIM | *NS* |
| PSM2PIM | *NS* | Code2PIM | *NS* |
| PSM2CIM | *NS* | Code2PSM | *S* |

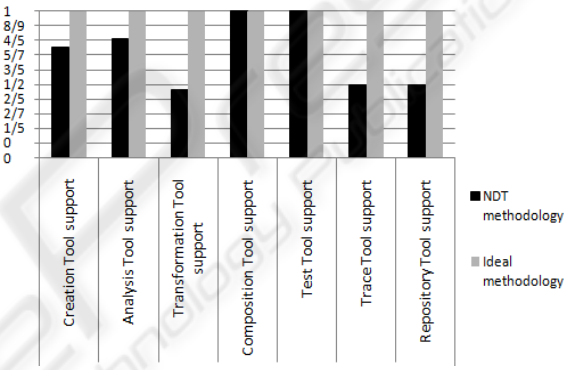


Figure 2: Graph of results in the Tool Support characteristic template.

## 5.2 A Reliability Quality Factor Evaluation of NDT Methodology for Tool Support

In the implementation in Microsoft Excel, Functionality, Reliability, Portability and Usability quality factors have been studied. In this example the Reliability factor is shown and its relations with the Tool Support characteristic. This is shown in Figure 3, where the black line represents Reliability on the NDT methodology and the grey line represents the ideal Reliability in an ideal approach according to the subcharacteristics under consideration. According to the results of the evaluation of the NDT methodology, only one characteristic has been considered in the example. Hence, the same subcharacteristics have been considered for each quality attribute of Reliability.

If other characteristics are considered (for example MDE characteristic) and subcharacteristics, then the results could be very different. In Figure 4, the Tool Support and the MDE characteristics are

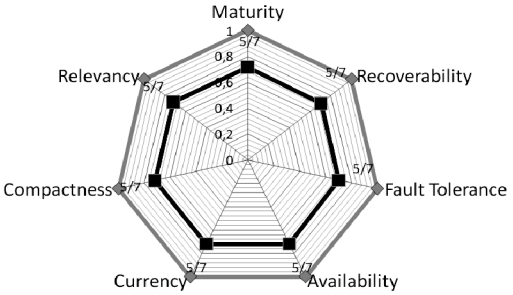


Figure 3: Reliability quality factor evaluation for the Tool Support characteristic.

considered for the evaluation of the Reliability quality factor about NDT methodology. In this case, is seen that NDT is better in Recoverability and Fault Tolerance than in Maturity, Relevancy, Compactness and Currency.

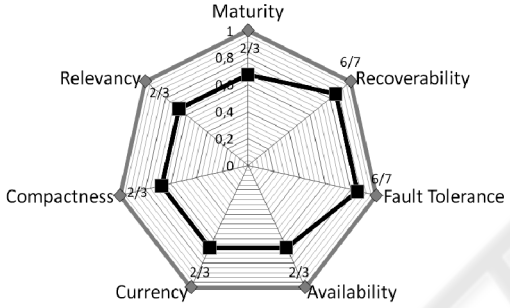


Figure 4: Reliability quality factor evaluation for Tool support and MDE characteristics.

# 6 CONCLUSIONS & FUTURE WORK

The fundamentals of a quality environment for MDWE methodologies are proposed in this paper. With regards to the contributions obtained from this research, a framework is required for the improvement of current proposals or the development of a new methodology. We consider that the use of QuEF will enhance the quality of products, processes and techniques of approaches. Furthermore, we have described those subcharacteristics related with the Tool Support characteristic which are required for the measurement of the value of MDWE methodologies in order to be able to assess and improve their Reliability. Therefore the use of QuEF can improve the efficiency and effectiveness of MDWE methodologies, and in turn may make their use more widespread since this approach evaluation helps one understand the strengths and weaknesses of a methodology. Further characteristics and quality factors have still to be developed. To this end, Microsoft Excel is currently being employed.

# ACKNOWLEDGEMENTS

This research has been supported by the project QSimTest (TIN2007-67843-C06_03) and by the RePRIS project of the Ministerio de Educación y Ciencia (TIN2007-30391-E), Spain.

# REFERENCES

C. Calero, J. Ruiz, M. Piattini. 2005, Classifying web metrics using the web quality model. Vol. 29 No. 3, pp. 227-248

F. J. Domínguez-Mayo, M. J. Escalona, M. Mejías, A. H. Torres. "Towards A Quality Evaluation Framework for Model-Driven Web Engineering Methodologies" WEBIST 2010. Vol. 2, pp. 191-194, 2010.

M. J. Escalona, G. Aragón, 2008. NDT. A Model-Driven Approach for Web Requirements. *IEEE Transactions on software engineering*, Vol. 34, No. 3.

M. J. Escalona, N. Koch, 2004, Requirements Engineering for Web Applications – A comparative study. Journal of Web Engineering. Vol. 2, No. 3, pp. 193-212.

IEEE Std 610.12-1990. *IEEE Standard Glossary of Software Engineering Terminology*,

ISO-International Organization for Standardization, ISO/IEC 9126-1, http://www.iso.org

C. Kroiβ, N. Koch, 2008. UWE Metamodel and Profile, User Guide and Reference. Technical Report 0802. Programming and Software Engineering Unit (PST), Institute for Informatics. Ludwig-Maximilians-Universität München, Germany.

P. Mohagheghi, V. Dehlen, 2008. Developing a Quality Framework for Model-Driven Engineering. MoDELS 2007 Workshops, LNCS 5002, pp. 275–286,

OMG: MDA. http://www.omg.org/mda/faq_mda.htm

W. Schwinger, W. Retschitzegger, A. Schauerhuber, G. Kappel, M. Wimmer, B. Pröll, C. Cachero Castro, S. Casteleyn, O. De Troyer, P. Fraternali, I. Garrigos, F. Garzotto, A. Ginige, G-J. Houben, N. Koch, N. Moreno, O. Pastor, P. Paolini, V. Pelechano Ferragud, G. Rossi, D. Schwabe, M. Tisi, A. Vallecillo, van der Sluijs and G. Zhang, 2008. A survey on web modeling approaches for ubiquitous web applications. *International Journal of web Information Systems Vol*. 4 No. 3, pp. 234-305.