LNCS 4607

# Web Engineering

7th International Conference, ICWE 2007
Como, Italy, July 2007
Proceedings

Lecture Notes in Computer Science 4607

*Commenced Publication in 1973*
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

## Editorial Board

Luciano Baresi Piero Fraternali
Geert-Jan Houben (Eds.)

# Web Engineering

7th International Conference, ICWE 2007
Como, Italy, July 16-20, 2007
Proceedings

Springer

Volume Editors

Luciano Baresi
Piero Fraternali
Politecnico di Milano
Dipartimento di Elettronica e Informazione
piazza Leonardo da Vinci 32, 20133 Milano, Italy
E-mail: {luciano.baresi, piero.fraternali}@polimi.it

Geert-Jan Houben
Vrije Universiteit Brussel
Department of Computer Science
Pleinlaan 2, 1050 Brussels, Belgium
Geert-Jan.Houben@vub.ac.be

Library of Congress Control Number: 2007930462

CR Subject Classification (1998): D.2, C.2, I.2.11, H.4, H.2, H.3, H.5, K.4, K.6

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-540-73596-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-73596-0 Springer Berlin Heidelberg New York



Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12089673 06/3180 5 4 3 2 1 0

# Table of Contents

## Services



## Metrics and Quality



## Caching

## Interfaces



## Models



## Verification and Testing



## Semantics and Web 2.0



## Search

## Application Development



## Demonstrations







## Industrial Track



## Doctoral Consortium

# Metamodeling the Quality of the Web Development Process' Intermediate Artifacts

Cristina Cachero[1], Coral Calero[2], and Geert Poels[3]

[1] University of Alicante. Spain
ccachero@dlsi.ua.es
[2] ALARCOS Research Group. University of Castilla-La Mancha. Spain
Coral.Calero@uclm.es
[3] Faculty of Economics and Business Administration, Ghent University. Belgium
geert.poels@ugent.be

**Abstract.** WE practices lack an impact on industry, partly due to a WE field that is not quality-aware. In fact, it is difficult to find WE methodologies that pay explicit attention to quality aspects. However, the use of a systematic process that includes quality concerns from the earliest stages of development can contribute to easing the building up of quality-guaranteed Web applications without drastically increasing development costs and time-to-market. In this kind of process, quality issues should be taken into account while developing each outgoing artifact, from the requirements model to the final application. . Also, quality models should be defined to evaluate the quality of intermediate WE artifacts and how it contributes to improving the quality of the deployed application. In order to tackle its construction while avoiding some of the most common problems that existing quality models suffer from, in this paper we propose a number of WE quality models to address the idiosyncrasies of the different stakeholders and WE software artifacts involved. Additionally, we propose that these WE quality models are supported by an ontology-based WE measurement meta-model that provides a set of concepts with clear semantics and relationships. This WE Quality Metamodel is one of the main contributions of this paper. Furthermore, we provide an example that illustrates how such a metamodel may drive the definition of a particular WE quality model.

**Keywords:** web quality process, web quality, web process quality, web measurement, ontology, metamodel, navigational model.

## 1 Introduction

It is an avowed fact that WE practices lack a big impact on industry [1]. This situation is at least partly caused by a WE field that is not quality-aware. In fact, it is difficult to find WE methodologies that include explicit support for quality aspects among their characteristics. In order to change this situation and assess the quality of the different WE artifacts, we need to define specific evaluation instruments. Such instruments may come in the shape of a *Quality Model,* defined by the ISO as the *set of characteristics and the relationships between them which provide the basis for specifying quality requirements and evaluating quality* [2].

L. Baresi, P. Fraternali, and G.-J. Houben (Eds.): ICWE 2007, LNCS 4607, pp. 74–89, 2007.




At this point, a question may arise: what should be the objective of such quality instruments? Garvin [3] defines five different quality perspectives, among which two have been widely adopted when defining quality models. The first one is 'quality as the degree of compliance with respect to certain specifications'. This is the most widespread perspective in Software Engineering due to the fact that in this kind of quality assurance the end user is not involved and therefore measures are easier and cheaper to take. The second perspective is quality as 'meeting customer needs'. Even if much more complex to evaluate, it is this second perspective the one that, according to the ISO/IEC 9126 [2] and ISO/IEC 14598 [4] standards, should make up the overall objective of any quality evaluation process. Provided that we narrow the term 'customer' to that of 'end-user', this concept of quality from the end-users' perspective is what the ISO/IEC 9126 standard defines as 'quality in use', that is, the *efficiency, productivity, security and satisfaction with which users use the application to satisfy specific goals under specific conditions.*

This ISO recommendation agrees with the fact that quality in use, even if it has not been tackled in a systematic way, is a widespread concern among Web developers, due to the necessity for most applications to keep the audience coming back to the site [5]. However, this concern is no reflected in current Web quality evaluation practices. If we examine the myriad of Web design guidelines [6] and automated measures [7] that can be gathered in literature we observe that, as it happens in Software Engineering, such Web evaluation effort reflects a 'conformance to specification perspective', that is, implicitly assumes the assessment of quality before the user is actually interacting with the application. One second problem of using such measures and guidelines is that, talking in terms of the OMG Standard Metapyramid [8] (see main subdivisions in Figure 1), the Web quality evaluation effort is concentrated on the M1-Implementation level (measures over the application code, without running it) and M0-test level of abstraction (code running under testing conditions). Only log analysis techniques have strived to evaluate the end-user actual behavior.

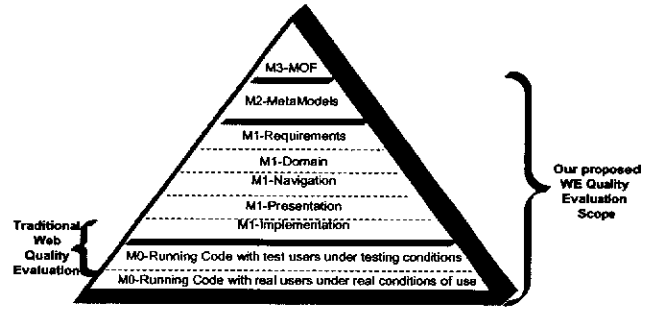


**Fig. 1.** The OMG standard metapyramid with additional WE subdivisions to distinguish among different levels of abstraction at M1 level (adapted from ISO10027)

Starting to assess quality at such a late stage of development is avowed to have a negative impact on the final product cost and quality [9]. In fact, according to Moody

and Shanks [10], the cost associated with removing a defect during design is on average 3.5 times greater than during requirements; at implementation stage the effort associated with removing the same defect can be up to 50 times greater, and up to 170 times greater after delivery. Other empirical studies have shown that moving quality evaluation effort up to the early phases of development can be 33 times more cost effective than testing done at the end of the development.

The solution to these two problems requires therefore to conciliate an early evaluation of the main internal Web products (that is, a 'conformance to specification' evaluation that is not just centered on code, but also makes use of early models, from requirements to implementation, see Figure 1) with the necessity of 'meeting customer needs', which implies taking into account the actual user behaviour under real conditions of use. Fortunately, the ISO set of quality standards establishes that the 'conformance to specifications' degree of a given software product (such as intermediate artifacts generated as part of a WE process) may be a valid predictor of the ability of the product to meeting user needs. This assumption means that is it possible to improve the Web quality in use by working on the quality of each outgoing artifact that participates in a typical WE process, from the requirements model to the final application to be delivered. The, only requisite is that the translation from the ' meeting user needs' (whose fulfillment is the final objective of the development process) quality perspective to the set of specific 'conformance to specifications' requirements defined for each model (which analysts/designers can systematically check) is accurately defined.

In order to perform such inclusion of quality concerns in existing WE methodologies in a sensible a consistent way, we have developed a proposal that has three main elements: (1) a quality-aware Web development process, (2) a set of general-purpose WE quality models specific for each stakeholder and/or WE artifact and, (3) a WE-Software Measurement Metamodel (SMM) that permits to operationalize and, if needed, also tailor, those quality models according to a particular domain and/or application. All three elements are based on principles and achievements that, uncovered in different quality lines of research, provide insights into how to deal with quality in each of the different workflows that a typical WE process defines, from requirements to implementation.

In this paper we are centering on the last two elements of our approach, that is, the definition of WE quality models and their adaptation to particular Web applications. To justify the necessity for our proposal, in Section 2 we present a brief overview of the main problems that existing quality models suffer from. How (1) the use of the Software Measurement Ontology (SMO) and (2) the definition of an associated WE measurement metamodel that guides the construction and adaptation of the quality model, can contribute to palliate such problems is presented in Section 3. This WE Measurement Metamodel must be instantiated to reflect a certain WE quality model as well as any necessary tailoring. An example of such instantiation that reflects an hypothetical Navigational quality model is developed in Section 4. Last, conclusions and future work are explained in Section 5.



## 2 Related Work

Quality models for software products are far from scarce. Well known pioneer models include McCall [12], Boehm[13], Dromey [14] and ISO/IEC 9126 [2]. All of them center on measurable elements over the implementation of the software product on one hand, and on the (abstract) quality characteristics on the other hand, and try to establish relationships among both dimensions.

There are various proposals of specific Web quality models, most of them tackling the Web idiosyncrasy from the 'meet the user needs' perspective [15, 16, 17, 18, 19]. From them, only [20] and [19] promote considering other artifacts (apart from code) that may take part in the WE development cycle, and none of them provide independent quality models for each level of abstraction. These approaches can however be refined and complemented by research in conceptual modeling quality (e.g. Lindland et a. framework [12], Krogstie et al. framework [21] and Moody and Shanks framework [11]), which provides further insight into how the quality concept can be dealt with at higher levels of abstraction. Last but not least, Web quality evaluation needs to be performed following a well defined quality evaluation process. Some well known Web quality evaluation processes are WebQUEM [15] and WebTango [7]. The main drawback of these processes is that they assume that Web quality evaluation is performed on the deployed application. Only [11] and [19] present a broader perspective and try to conciliate Web quality evaluation with a general WE development process, even if they only provide general guidelines and not specific proposals. Next, we present the challenges all these fields pose, and how we propose to integrate them in a single, consolidated proposal in the context of WE.

### 2.1 Research Issues

When trying to operationalize all the myriad of different quality models and quality evaluation processes that have been proposed in literature, several theoretical and practical issues arise: [10, 22]:

- **P1: Terminology inconsistencies.** Most approaches (the exception being those based on theoretical grounds) lack a definition for quality concepts that is precise and concise. For instance, while in the ISO/IEC 9241-11 usability refers to the end-user perception as a whole (and therefore encompasses efficiency effectiveness and satisfaction), in the ISO/IEC 9126 end-user perception is referred to as 'quality in use', and usability is only one of the internal characteristics that may affect such quality in use.
- **P2: Partially defined.** Most quality models are outlined but not fully developed. All define measurable concepts, some of them also attributes, few of them include (most often partial) measures and scarcely any defines decision criteria or indicators.
- **P3: Lack of focus.** Most quality models provide an extensive (and mostly tangled) coverage of stakeholders and levels of abstraction. An example of such assertion is the QUIM model [22], which aims at being a consolidated usability model that integrates all possible perspectives. As another example, WQM [16] covers 10 factors, 26 subfactors and 127 measures that may be related to any WE artifact, from analysis to implementation.

- **P4: Disregard for process quality.** Most quality models define criteria and, in some cases, measures for evaluating products (error detection), but not how to develop products in a way that assures a certain level of quality (error prevention).
- **P5: Lack of integration with current practices.** Quality management is not integrated into current WE practices.
- **P6: Lack of simplification and validation.** Quality models that include measures usually pay little attention to the theoretical/empirical validation of the included measures. Furthermore, although empirical research has shown that a few measures most times suffice to obtain significant gains in quality ([11]), quality models usually include an extensive, even redundant set of measures. Such verboseness unnecessarily increases the complexity and therefore hampers the potential usefulness of the quality models.
- **P7: Interdependencies and measure interpretations not clear.** In most quality models (again the notable exception being those that are based on theory), the degree of influence of individual internal quality factors on the quality in use of the application, as well as their interdependencies, are not well established. For example, the role of learnability versus understandability in the usability model presented in [19] is an open issue. Also, little information is provided on how to interpret measurement results.
- **P8: Lack of tool support.** Although most Web measures are automated, tool support for the definition of quality models and, even more important, for the automation of the measurement process on a given Web application is still an open issue.
- **P9: Lack of guidelines for improvements.** Even in the case of being able to evaluate a certain Web characteristic, to our knowledge extent no quality model provides a clue about how (by means of which changes in the artifacts) such evaluation could be improved, let alone to which extent such changes may affect the evaluation of other characteristic included in the quality model.

In order to overcome these problems, certain requirements should be achieved when defining WE quality models and integrating them with WE development processes:

**Requirement 1.** WE quality models should be expressed using a set of clear concepts with clear semantics and relationships, in order to ease their understanding and assure a structural coherence. This palliates problems P1, P2 and P3.
**Requirement 2.** WE quality models should be defined taking into account a specific stakeholder and a specific software artifact. This palliates problem P3.
**Requirement 3.** WE quality models should be accompanied by a WE quality evaluation process. Such process must be defined and integrated with the WE development process. This contributes to overcome problems P3, P4, and P5. Furthermore, for the definition of the WE quality evaluation process, standards should be followed when possible. This alleviates problem P6.
**Requirement 4.** Guidelines should be provided when possible to improve WE artifacts according to the WE Quality Model under consideration. Such guidelines should also if possible preserve the semi-automatic nature of the WE process. This contributes to solve problem P9.
**Requirement 5.** The integration of WE quality models in the WE process should preserve the semi-automated nature of such process. This contributes to alleviating problem P8.



**Requirement 6.** WE quality models should be empirically validated before being included in the WE process. This palliates problems P6 and P7.

In this paper, we will center on how the definition of WE Quality models that preserve *Requirement 1* and *Requirement 2* can be achieved if we base them on an ontology-based WE measurement meta-model. The use of a meta-model assures the syntactic correctness of the WE Quality Models (including completeness restrictions and focus control on specific stakeholders and specific WE artifacts), while the fact that this meta-model is based on an ontology contributes to avoiding terminology inconsistencies.

## 3 Definition of WE Quality Models Following an Ontology and a Meta-model

As we presented previously, one of the problems that existing quality models face is terminology inconsistencies. In order to overcome such problem we need a common vocabulary both to express WE concepts and to express quality concepts. Such common vocabulary usually comes in ontology form.

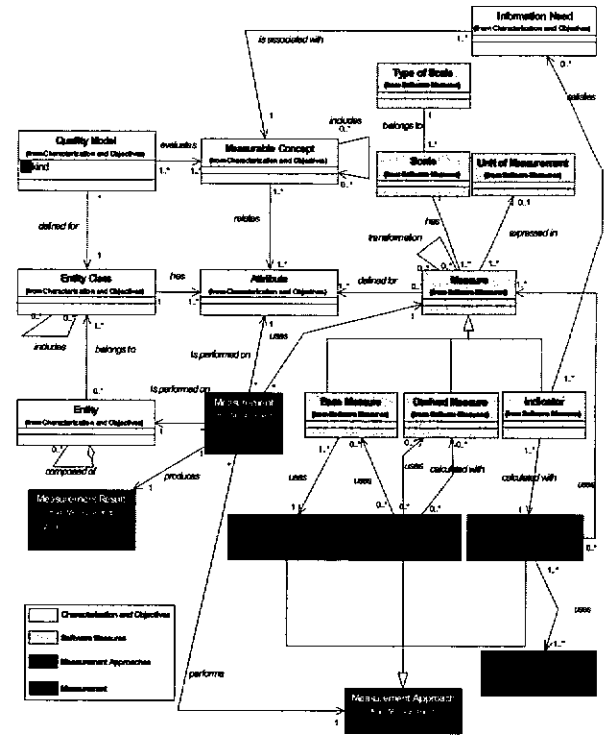


**Fig. 2.** The UML diagram of the SMO

Ontologies, defined as explicit, formal and shared specifications of a conceptualization, have been widely used in Software Engineering [23]. The use of an ontology not only avoids vocabulary conflicts and inconsistencies but also establishes the adequate level of detail for the definition of each concept.

While the definition of a WE ontology is still being worked on and remains out of the scope of this paper, the greater maturity of the measurement field causes some proposals for measurement ontologies to co-exist. From them, the Software Measurement Ontology (SMO)[24] is, to our knowledge extent, the most complete one, which is the reason why we have chosen it as the basis for our approach. The SMO ontology is structured around four packages, namely:

- **Software Measurement Characterization and Objectives**, which includes the concepts required to establish the scope and objectives of the software measurement process.
- **Software Measures**, which aims at establishing and clarifying the key elements in the definition of a software measure.
- **Measurement Approaches**, which introduces the concepts necessary for reflecting measurement results.
- **Measurement**, which establishes the terminology related to the act of measuring software.

**Table 1.** SMO terms definition

| Term | Definition |
|---|---|
| Measurement Approach | Sequence of operations aimed at determining the value of a measurement result. (A measurement approach is either a measurement method, a measurement function or an analysis model) |
| Measurement | A set of operations having the object of determining the value of a measurement result, for a given attribute of an entity, using a measurement approach |
| Measurement Result | The number or category assigned to an attribute of an entity by making a measurement |
| Information Need | Insight necessary to manage objectives, goals, risks, and problems |
| Measurable Concept | Abstract relationship between attributes of entities and information needs |
| Entity | Object that is to be characterized by measuring its attributes |
| Entity Class | The collection of all entities that satisfy a given predicate |
| Attribute | A measurable physical or abstract property of an entity, that is shared by all the entities of an entity class |
| Quality Model | The set of measurable concepts and the relationships between them which provide the basis for specifying quality requirements and evaluating the quality of the entities of a given entity class |
| Measure | The defined measurement approach and the measurement scale. (A measurement approach is either a measurement method, a measurement function or an analysis model) |
| Scale | A set of values with defined properties |
| Type of Scale | The nature of the relationship between values on the scale |
| Unit of Measurement | Particular quantity, defined and adopted by convention, with which other quantities of the same kind are compared in order to express their magnitude relative to that quantity |
| Base Measure | A measure of an attribute that does not depend upon any other measure, and whose measurement approach is a measurement method |
| Derived Measure | A measure that is derived from other base or derived measures, using a measurement function as measurement approach |
| Indicator | A measure that is derived from other measures using an analysis model as measurement approach |
| Measurement Method | Logical sequence of operations, described generically, used in quantifying an attribute with respect to a specified scale. (A measurement method is the measurement approach that defines a base measure) |
| Measurement Function | An algorithm or calculation performed to combine two or more base or derived measures. (A measurement function is the measurement approach that defines a derived measure) |
| Analysis Model | Algorithm or calculation combining one or more measures with associated decision criteria. (An analysis model is the measurement approach that defines an indicator) |
| Decision Criteria | Thresholds, targets, or patterns used to determine the need for action or further investigation, or to describe the level of confidence in a given result |



In Figure 2 the UML diagram of the ontology is presented, while in Table 1 the concepts defined in the ontology are shown.

This ontology is the basis on which a namesake (and structure-equivalent) metamodel has been defined [25]. Next, we present how we have adapted such metamodel to meet our detected needs.

### 3.1 The WE Software Measurement Meta-Model (WE-SMM)

The Software Measurement Meta-Model (SMM) presented in [25] is a mirror of the Software Measurement Ontology presented in Figure 2, and may be instantiated to define in a systematic and non-ambiguous way a quality model that includes all the necessary concepts for its operationalization. The main advantage of using metamodels instead of ontologies in the context of a software development process stems in their prescriptive rather than descriptive nature, what permits the designer to make assumptions on the quality models that are not possible with ontologies. Furthermore, while ontologies need to be general, meta-models can be tailored to meeting specific needs. In our case, and given the fact that we aim at simplifying as much as possible the definition of WE Quality Models, we have adapted the SMM to the WE environment, with the aim of making its instantiation more intuitive for Web designers. Such WE-SMM is presented in Figure 3.

Summarizing, the construction of this WE-SMM has implied the following actions over the original SMM:

- We have limited the risk for inconsistencies in the measurement model by eliminating SMM redundant relationships.
- We have limited the set of valid Entity Classes to the outgoing artifacts of the WE development process. In this way, measurable concepts that are to be measured on different WE artifacts are forced to belong to different quality models.
- We have introduced a global Information Need that is connected with the WE-quality model as a whole to justify its definition. For the structure of this Global Information Need we propose to use the GQM template for goal definition [13].
- In order to keep the quality model simple, we have restricted the number of Information Need objects that can be associated wich each Measurable Concept (1)
- For the same reason, we have established that each Information Need be satisfied by a single Indicator, implying that the Measurable Concept connected with the Information Need is also (transitively) associated with that indicator.
- In order to assure that every Attribute is measurable, every attribute defined in a WE quality model should be associated with at least one Measure that is devoted to measuring such Attribute. This restriction makes sure that the evaluation model is operationally defined by means of Measures, that is, no reliant on subjective interpretations of concepts [10].
- In order to further contextualize the WE quality model and help to keep the focus, we have added a 'Stakeholder' element to the original SMM. Stakeholders are usually not explicitly identified in existing quality models. However, as stated in [11], they are important in any quality model, as different Stakeholders will generally be interested in different Measurable Concepts. The instantiation possibilities of this

concept in the context of WE are (1) Analysts/Designers, (2) Developers/Maintainers and (3) Customers (subdivided into Acquirers and End-Users)
- Finally, we have omitted from the WE-SMM the Measurement package, due to the fact that their elements do not contribute to the definition of quality models but rather to the results of their operationalization.

Additionally, and although not directly reflected in the WE-SMM, in order to control the quality model complexity we recommend the limitation of the hierarchy depth of Measurable Concepts to two levels of detail. Also, following the ISO/IEC 9126 example, these two levels should be characterized by familiar labels and concise definitions. Similarly, attributes associated with Entity Classes should also be familiar and provide concise definitions. Finally, in order to facilitate a hypothetical merging of quality models at different levels of abstraction into a general, well-structured WE Global Quality Model, we recommend that attributes for the different models have unique names in the context of the WE field.

The concepts and relationships included in this meta-model, together with the additional recommendations, force a certain structure similarity among any quality model

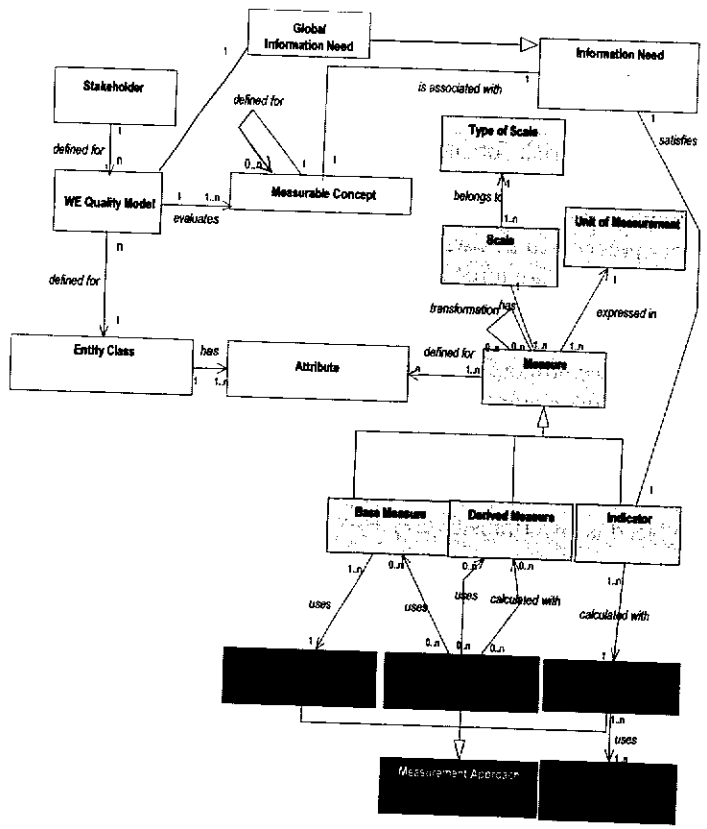


**Fig. 3.** WE-Measurement metamodel



defined based on it, what in turn facilitates the understanding and discussion of WE quality models among both researchers and practitioners.

With a WE-SMM that is based on a SMO we are fulfilling Requirements 1 and 2 (see Section 2). Next we present an example of how to use these instruments to define particular WE Quality Models.

## 4 WE-SMM Based Definition of WE-Quality Models

The WE-SMM defined above can be easily instantiated to define a complete and structurally sound quality model over any WE intermediate artifact with the ultimate objective of assuring certain characteristics that may contribute to improving the quality in use of the application. Such quality model can be general (when it does not take into account the particularities of the domain and/or application under evaluation) or particular, if such knowledge introduces any divergence in the elements that are taken into account for the evaluation task, This means that, while Quality Models in literature usually reflect a global view of the set of measurable concepts, measures and so on that may be applied at a certain level of abstraction (that required by the measures included), our WE-SMM instantiation may furthermore involve a tailoring process over the original Quality Model to adapt it to a given application in a given domain. During this tailoring process certain fine-tuning actions can be performed. For instance, more specific decision criteria that better reflect the domain knowledge could be defined, or certain attributes/measures may be dismissed to even further simplify the measurement process. Furthermore, we say that the WE-SMM instantiation operationalizes a Quality Model, due to the fact that (1) it obligues the QM to present certain characteristics (e.g. to provide measures for every attribute) and (2) it permits to express the Quality Model it in a machine-readable format, which in turn opens the path to applying automation techniques.

In order to illustrate this fact, next we are presenting an instantiation example that operationalizes a hypotetical quality model devoted to evaluating the navigability of a Web application. Although the whole definition of this WE Quality Model is out of the scope of this paper, for illustration purposes let's assume that this Navigability WE-QM includes an Understandability characteristic that can be measured based on two attributes: Navigation Node Complexity and Navigation Path Complexity. The definition of the meaning of these concepts was presented in Table 1.

This measurement model intends to reflect the viewpoint of the end-user of the application, that is, the **Stakeholder** involved is the End-User. Therefore, only model qualities that are bound to contribute to increasing the end-user quality in use of the application should be included in this instantiation. Such instantiation aims at assessing the navigability problems that may arise due to a low-quality definition of navigational paths and navigational nodes.

Due to the fact that navigation paths and nodes are both defined by means of a navigation model, the **Entity Class** will be such Navigation Model, which is part of any WE methodology.

The WE-Quality Model is associated to a **Global Information Need** To Know how good Navigability is. Recall that the description of such global information need

must follow the GQM template, and therefore could be defined as follows: analyzing the WE Navigational Model for the purpose of evaluating it with respect to the navigability of the final application from the viewpoint of the end-user of the application in the context of a testing environment.

The **Navigability WE-Quality Model** contains a set of **Measurable Concepts**. If we consider Navigability as 'Usability of the navigation', we can assume that the main characteristics included in the ISO 9126-1 for Usability apply. These characteristics are Understandability, Learnability, Operability, Attractiveness and Compliance. We agree with [19] in that the first three Measurable Concepts (understandability, learnability, operability) are related with the user performance and can be therefore quantified using objective measures, some of which can be taken over navigational models. Attractiveness is not relevant at this stage of development, where final users are not yet present. Last, as far as we know there are no widely accepted standards or conventions regarding the definition of navigation structures in WE navigational models, and therefore Compliance is not relevant either.

**Table 2.** WE-Measurement Meta-model instantiation example

| Term | Instantiation for the Understandability Measurable Concept |
|---|---|
| Stakeholder | End-User |
| Global Information Need | To know how good navigability is |
| Information Need | To know how good understandability is |
| Measurable Concept | Understandability |
| Entity Class | Navigation model |
| Attribute | (1) Navigation Node Complexity – (2) Navigation Path Complexity |
| WE-Quality Model | Navigability WE-Quality Model |
| Base Measure | (1) Number of attributes (NA) - (2) Number of Navigational Links (NNL) |
| Scale | Natural Number |
| Type of Scale | Ratio |
| Measurement Method | (1) Count the number of attributes of the model– (2) Count the number of links of the model |
| Indicator | UND_IND (NA, NNL) |
| Scale | Acceptable-NonAcceptable |
| Type of Scale | Ordinal |
| Analysis Model | f(UND_IND)=NA+NNL |
| Decision Criteria | If f(UND_IND) <50 then Acceptable else NonAcceptable |

Each one of these Measurable Concepts must be related to an **Information Need**. The Information Need covered by Understandability in the context of the Navigability WE-Quality Model is To Know how good Understandability is. The description associated with such concept could be 'the capability of the Web application navigational structure to enable the user to understand whether the application is suitable for her, and how it can be used for particular tasks under certain conditions of use'. Learnability and Operability can be defined similarly.

The next step consists in defining the **attributes** that may influence each characteristic. The navigational model has two main purposes in the WE development process. On one hand it defines the set of abstract pages, that is, the basic information nodes that make up the application. On the other hand, it provides the navigation paths and the navigation facilitators (menus, indexes, guided tours and so on) to improve the user experience. With these two purposes in mind, we have identified two Navigation Model Attributes:



1. Navigation Node Complexity: possibly related to learnability, and understandability. Such relationship is not in the meta-model because it is derived from the relationship between measures-attributes on one hand and the empirically validated relationship measure- measurable concepts on the other hand.
2. Navigation Path Complexity: possibly related to learnability, understandability and operability. Again, such relationship must be empirically validated (it would be a derived relationship)

The partial instantiation of the WE-SMM that gathers all the concepts presented so far is presented in Figure 4.

According to the meta-model, each model attribute must be related to at least one **measure**. For the sake of the example, let's suppose that we have determined that only two measures are relevant for the evaluation purposes of this Navigability WE-Quality Model: the number of navigational links (NNL) and the number of attributes (NA).

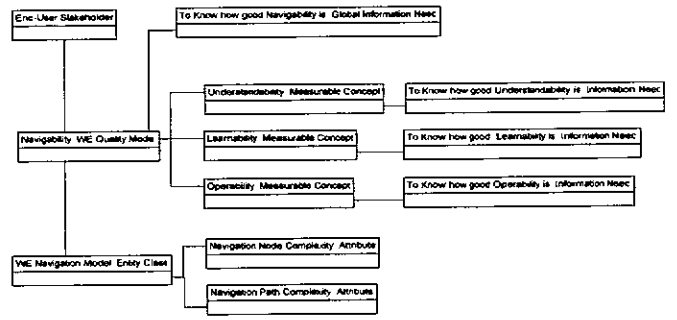


**Fig. 4.** Partial instantiation of WE_Navigability model (Part 1)

The definition of the Number of Navigational Links (NNL) **Base Measure** includes the **Scale** Natural Number, the **Type of Scale** Ratio and the **Unit of Measurement** Links. This measure is associated with the Navigation Path Complexity Attribute. The **measurement method** is 'to count the number of links of the model'. Similarly, the definition of the Number of Attributes (NA) Base Measure is associated with the Scale Natural Number, the Type of Scale Ratio and the Unit of Measurement Attributes. This measure is related with the Navigation Node Complexity Attribute. The **measurement method** is 'to count the number of attributes of the model'.

Also, each information need requires at least one **Indicator**. Indicators can be regarded as special kinds of measures that are related to decision criteria via an **Analysis Model**.

As an example, let's define the Understandability Indicator UND_IND. Let's suppose that the Analysis Model associated to this indicator is a function that involves the two measures presented above: F(UND_IND)=NNL+NA.

Let's also assume that this indicator belongs to the Scale {Acceptable, Non Acceptable} and Type of Scale Ordinal (Acceptable is better than Non Acceptable).

Last, for the definition of the decision criteria let's assume that the Trellis number applies, and that models with less than 50 elements are understandable enough. This

decision criteria is expressed in the meta-model instantiation as if f(UND_IND) <50 then Acceptable else NonAcceptable.

Figure 5 presents a WE-SMM instantiation that reflects all these new elements of the Navigability WE-Quality Model.

As the reader may have already noticed, this measurement model is quite straightforward to use by any designer familiar with Navigation Models. The fact that we specifically consider the stakeholder helps to focus the measurable concepts, attributes and measures that must be taken in consideration.

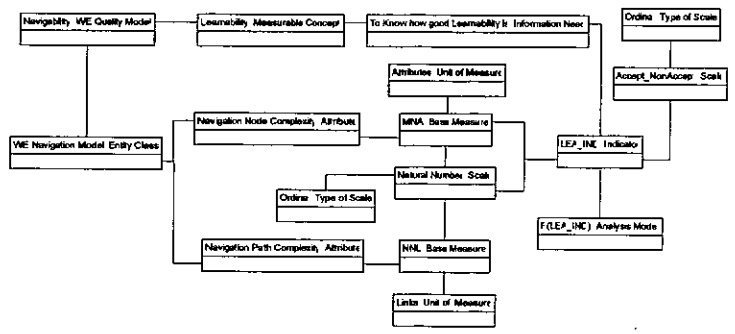


**Fig. 5.** Partial instantiation of WE_Navigability Model (Part 2)

## 5 Conclusions and Future Work

The systematic integration of quality issues in the WE field is mandatory if we aim at setting the focus on preventing rather than detecting errors and therefore decreasing maintainability costs. Even more important, we believe that providing practitioners with WE methodologies that assure a certain degree of quality of the application delivered is likely not only to support some of the WE traditional claims of providing better results than creative practices, but also to increase the acceptance rate of the WE technology in industry.

These quality issues should be taken into account while developing each outgoing artifact, from the requirements model to the final application to be delivered. In order to perform such inclusion of quality concerns in existing WE methodologies in a sensible and consistent way, we have developed a proposal that enriches the traditional WE development process with a set of quality activities and instruments that serve to evaluate the quality of intermediate WE artifacts as a means to improving the quality of the deployed application. Our proposal has three main elements: (1) a quality-aware Web development process (out of the scope of this paper), (2) a set of general-purpose WE quality models specific for each stakeholder and/or WE artifact and, (3) a WE software measurement metamodel (SMM) that permits to operationalize and, if needed, also tailor, those quality models according to a particular domain and/or application. All three elements are based on principles and achievements that, uncovered in different quality lines of research, provide insights into how to deal with quality in each of the different workflows that a typical WE process defines, from requirements to implementation.



This paper presents two contributions. On one hand, it provides an overview of the main problems associated with quality models in Software Engineering, and a set of requirements that should be met by any quality model proposal if it aims at being of use. On the other hand, we clarified how some of these requirements can be covered by defining quality models with certain elements and restrictions. In order to assure this, we have defined a WE-SMM that tailors an existing SMM. This WE-SMM and the way we propose to use it to instantiate sound and complete WE quality models is the second contribution of this paper.

If we review the list of problems presented in Section 2, the fact that such WE-SMM is based on an underlying ontology contributes to avoiding terminology inconsistencies (P1). Also, the use of this metamodel turns the descriptive nature of ontologies into prescriptive, and therefore assures that a set of syntactic and semantic constraints are met by any quality model defined as an instantiation of such meta-model. One of such constraints is the set of elements that must be present in any syntactically correct WE quality model, which partially solves P2. The focus on a given field (in our case the WE field) as the application context of the quality models facilitates the task of constructing consolidated, exhaustive yet specialized models. This fact also contributes to alleviating P2 and P3. Additionally, the consideration of the WE development process with its related stakeholders and outgoing artifacts allows us to univocally define (1) the particular stakeholder at which each WE quality model is aimed and (2) the measurable concepts, attributes and measures that are applicable to each specific artifact. This fact also contributes to improving P3. Last but not least, this way of representing WE quality models by means of a meta-model instantiation is a machine-readable way, which helps to preserve the development advantages provided by the (semi-)automatic nature of WE processes. This in turn leverages P8. As a proof of concept we have presented a navigational quality model by using the meta-model, together with an indicator. This quality model proposal must be validated empirically in order to determine if the selected indicator and decision criteria are valid (which is one of the requirements presented in Section 2 that are not covered in this work). In fact, this is one of the main further lines of research: defining **empirically validated quality models** that include all the elements referenced in the WE-SMM for each level of abstraction is far from easy, yet it is an unavoidable step if we eventually aim at assuring that the results of the WE quality evaluation process are trustworthy. Another important future line of research is how the improvement of the intermediate models based on the quality evaluation results should be tackled in order to actually assure a good quality of the final product.

## Acknowledgments

This paper has been supported by the MEC Spanish Ministry for Staff Stages at foreign Universities (PR2006-0374) and projects CALIPSO (TIN20005-24055-E), MEC-FEDER (TIN2004-03145), ESFINGE (TIN2006-15175-C05-05), METASIGN (TIN2004-00779) and DSDM (TIN2005-25866-E)). Also, is part of the DADASMECA project (GV05/220), financed by the Valencia Government and DIMENSIONS (PBC-05-012-1) financed by the Castilla-La Mancha government.

## References


[1] Lang, M., Fitzgerald, B.: Hypermedia Systems Development Practices: A Survey. IEEE Software 22(2), 68–75 (2005)

[2] ISO/IEC 9126. Software engineering – Product quality – Part 1: Quality model. International Organization for Standardization, Geneva (2001)

[3] Garvin, D.: What Does "Product Quality" Really Mean?, Sloan Management Review, Fall 1984, pp. 25–45 (1984)

[4] ISO/IEC 14598. Information technology – Software Product Evaluation. International Organization for Standardization. Geneva (1999)

[5] Fraternali, P., Paolini, P.: Model-driven development of Web applications: the Autoweb System. ACM Transactions on Information Systems (TOIS) 18(4), 323–382 (2000)

[6] Nielsen, J.: Designing Web Usability: The Practice of Simplicity. New Riders, Berkeley (2000)

[7] Ivory, M.Y.: Automated Web Site Evaluation. Kluwer Academic Publishers, Norwell (2004)

[8] ISO/IEC 10027. Information technology – Information Resource Framework. International Organization for Standardization. Geneva (1990)

[9] Briand, L., Morasca, S., Basili, V.: Defining and Validating Measures for Object-based High-level Design. IEEE Transactions on Software Engineering 25(5), 722–743 (1999)

[10] 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)

[11] Moody, D.L., Shanks, G.G.: Improving the quality of data models: empirical validation of a quality management framework. Information Systems, vol. 28, pp. 619–650. Elsevier Science Ltd (2003)

[12] McCall, J.A., Richards, P.K., Walters, G.F.: Factors in Software Quality, Nat'l Tech. Information Service, vol. 1, 2 and 3 (1977)

[13] Bohem, B.W.: Software Engineering Economics. Prentice Hall Inc., Englewood Cliffs, NJ (1981)

[14] Dromey, R.G.: A model for software product quality. IEEE Transactions on Software Engineering 21(2), 146–162 (1995)

[15] Olsina, L., Rossi, G.: Measuring Web Application Quality with WebQEM. IEEE Multimedia Magazine 9(4), 20–29 (2002)

[16] Calero, C., Ruiz, J., Piatinni, M.: A Web Metrics Survey Using WQM. In: Koch, N., Fraternali, P., Wirsing, M. (eds.) ICWE 2004. LNCS, vol. 3140, pp. 147–160. Springer, Heidelberg (2004)

[17] Moraga, M., Calero, C., Piattini, M.: Ontology Driven Definition of a Usability Model for Second Generation Portals. In: MATES 2006, vol. 155, ACM Press, NewYork (2006)

[18] Comai, S., Matera, M., Maurino, A.: A Model and an XSL Framework for Analyzing the Quality of WebML Conceptual Schemas. In: Olivé, À., Yoshikawa, M., Yu, E.S.K. (eds.) Advanced Conceptual Modeling Techniques. LNCS, vol. 2784, pp. 339–350. Springer, Heidelberg (2003)

[19] Abrahao, S., Insfran, E.: Early usability evaluation in Model-Driven Architecture Environments. In: Proceedings of the Sixth IEEE International Conference on Quality Software. IEEE Press, Wiley, Chichester (2006)

[20] Comai, S., Matera, M., Maurino, A.: A Model and an XSL Framework for Analyzing the Quality of WebML Conceptual Schemas. In: Olivé, À., Yoshikawa, M., Yu, E.S.K. (eds.) Advanced Conceptual Modeling Techniques. LNCS, vol. 2784, pp. 339–350. Springer, Heidelberg (2003)

[21] Krogstie, J., Lindland, O.I., Sindre, G.: Defining quality aspects for conceptual models. ISCO 1995: 216-231 (1995)





[22] Seffah, A., Donyaee, M., Kline, R.B., Padda, H.K.: Usability measurement and metrics: a consolidated model. Software Quality Journal 14, 159–178 (2006)

[23] Ruiz, F., Hilera, J.R.: Using Ontologies in Software Engineering and Technology. In: Calero, C., Ruiz, F., Piattini, M. (eds.) Ontologies for Software Engineering and Software Technology, Springer, Heidelberg (2006)

[24] García, F., Bertoa, M., Calero, C., Vallecillo, A., Ruiz, F., Piattini, M., Genero, M.: Towards a consistent terminology for software measurement. Information and Software Technology 48(8), 631–644 (2005)

[25] Ferreira, M., García, F., Bertoa, M., Calero, C., Vallecillo, A., Ruiz, F., Piattini, M., Braga, J.L.: Medición del Software: Ontología y Metamodelo. Technical Report, University of Castilla-La Mancha (2006)