Towards a Quality-Aware Web Engineering Process

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

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

Abstract. Evidence-Based Web Engineering (WE) is necessary in order to (1) help industry practitioners in making rational decisions about technology adoption and (2) increase the acceptability of WE methodologies. Particularly, empirical data should be provided to support traditional WE claims such as increased productivity or better quality of the applications deployed using a WE methodology. Unfortunately the WE community is not yet familiar with either systematic quality evaluation issues or empirical research, and therefore tools and guidelines to ease this shift are necessary. In this paper we extend the traditional WE Development Process with quality evaluation and assurance activities that are compliant with the ISO/IEC 14598 set of standards and guarantee that Web applications developed with WE approaches fulfill certain quality criteria. This extension follows the MDA paradigm in order to ensure that the development productivity is not hampered by the additional focus on quality aspects.

Keywords: Web Engineering methodologies, Web Engineering process, quality evaluation, quality assurance, model-driven development

1 Introduction

It is an avowed fact that WE practices lack an impact on industry [14]. This situation is at least partly caused by a WE field that does not guarantee any kind of improvement over ad-hoc approaches towards assuring the quality in use of the deployed applications, where by 'quality in use' we mean the *efficiency, productivity, security and satisfaction with which users use the application to satisfy specific goals under specific conditions* [10]. In fact, WE methodologies and their associated development processes do not usually include specific support for the specification and implementation of quality requirements. This fact contrasts with the definition of WE as "the application of systematic, disciplined and **quantifiable** approaches to the cost-effective development and evolution of high-quality applications in the World Wide Web" [9]. One possible reason for this situation is that, being the final objective of any quality evaluation process the quality in use (meeting user needs) [10], and given the fact that assessing quality in use means tracking the use that real users make of the application under real exploitation conditions, the WE field has traditionally considered such concerns out of its scope. However, this consideration disregards an important relationship between quality in use and other quality perspectives. Namely, according to ISO/IEC 9126, there is a relationship between quality as 'meeting user needs' (that is, quality in use) and quality as a 'conformance to specifications' [8]. Otherwise stated, it is possible to predict the degree of quality in use of the final Web application by examining which quality specifications are met by the intermediate products (Web artifacts) of a certain application. The advantages of such change of perspective translate into cost and quality gains [3].

The scene regarding quality evaluation in the context of the traditional ad-hoc approaches towards Web site/application development is just slightly better. Here, the concern for Web quality is showed on the wide range of Web design guidelines and automated measures that can be gathered in literature [12, 18]. Additionally, and due to the high expense associated with monitoring the use of the application under real conditions of use, most of these guidelines and measures (the notable exception being those based on log analysis techniques) reflect a 'conformance to specification perspective'. In these approaches, the relationship between specifications and user needs is usually implicitly assumed, without providing empirical evidence. Also as a drawback, the lack of intermediate models supporting those ad-hoc approaches causes measures to still be centered on the lower (mostly code) levels of abstraction. Given these facts, it seems clear that an increase of the level of abstraction at which Web guidelines/measures are applied would be desirable, and such change can only be achieved if WE practices are adopted and if the WE process includes a quality evaluation and assurance perspective, from the early development stages till deployment.

In this paper we present a quality-aware WE process that fulfills these conditions and systematically integrates quality evaluation and assurance issues at every stage of the WE process without hampering the cost and/or time to market of the delivered application. Our proposal explicitly recognizes the relationship between the 'meeting user needs' and 'conformance to specifications' perspectives, and provides a working basis to empirically prove such relationships. In order to perform such inclusion of quality concerns in existing WE methodologies in a sensible, consistent and practical way, our research aims at the development of three elements: (1) a quality-aware WE 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 (WE-SMM) that permits the developer to operationalize and, if needed, also tailor, those quality models according to a particular domain and/or application. In this paper we are centering on the first element (the process). Readers interested in the quality instruments (Quality Models and the WE-SMM) that support such processes are referred to [5]. To justify the necessity for our proposal, in Section 2 we present a brief overview of the state of the art in quality instruments and WE quality evaluation processes. In Section 3 the generic WE process is presented, together with an analysis of how each artifact and/or activity may influence the quality in use of the deployed application. Section 4 then explains our proposal of a quality-aware generic WE process and how it was developed based on the ISO 14598 set of standards [11]. Last, conclusions and future work are presented in Section 5.

2 Related Work

In order to evaluate quality it is necessary to count on instruments that are based on clear definitions. One of these instruments is a **quality model**. A quality model is defined in ISO as the *set of characteristics and the relationships between them which provide the basis for specifying quality requirements and evaluating quality*. Quality models for software products are far from scarce. Well known pioneer models include the McCall, Boehm and Dromey models. Because of the widespread use of the ISO family of quality-related standards (including the ISO/IEC 9126 software quality standard), many proposals have aimed at tailoring/refining/improving these standards. For example, the Quint2 quality model regards the ISO/IEC 9126 standard as a valid but incomplete quality model, and therefore tries to complete it with additional features. If we now focus on WE, there are various proposals of specific WE quality models, most of them tackling the specificities of the 'meet the user needs' quality perspective on Web applications [1, 6, 7, 17, 19]. From these proposals, only [1] and [7] consider the quality of other artifacts than code that are constructed during the WE development cycle, and none of them provides independent quality models for the different levels of abstraction in the WE process. These approaches can however be refined and complemented by research on model quality (e.g. Lindland et a. framework [15], Krogstie et al. framework [13] and Moody and Shanks framework [16]), which provides further insight into how the quality concept can be dealt with at higher levels of abstraction than the actual software code.

Another instrument is a **quality evaluation process** that prescribes *how and when* quality evaluation must be performed. An example here is the ISO/IEC 14598 series of standards that gives an overview of software product evaluation processes and provides guidance and requirements for evaluation of general software products. In the WE field, some quality evaluation processes are WebQUEM [19] and WebTango [12]. Similar to the proposed WE quality models, the main drawback of most of these processes is that they assume that the quality evaluation is performed on the deployed application. In fact, only [17] and [1] recognize the necessity to conciliate the WE quality evaluation process with the general WE development process. The general structure of such WE development process is presented next.

3 The WE Process

The relative immatureness of the WE field causes a lack of agreement on a common Web development process. However, most methodologies share a set of artifacts and activities that are presented in Figure 1 and that may be regarded as a simplified WE process where, for the sake of readability, the iterative and incremental nature of this process has been hidden. This process, based on the Model Driven Engineering paradigm, departs from a general business model and includes (1) a (manually performed) functional requirements workflow, whose outgoing artifact is a use case model, (2) an analysis workflow, whose output is a domain model (usually an ER diagram or a UML class diagram), (3) a conceptual design workflow, whose outputs are a navigation and a presentation model (expressed by means of UML profiles or proprietary), (4) a detailed design workflow that introduces platform and technology specific features (typically J2EE and .NET) and (5) an implementation workflow, which results in a Web application that is ready to be deployed. Variants of this process model exist, usually to include additional Platform Independent Models (PIM's) and/or Platform Specific Models (PSM's) (architectural models, business process models, different languages and/or platforms, etc.) that further enrich the application specification. Additionally, WE methodologies promote the use of automatic and/or (semi-)automatic transformations among most of these artifacts (depicted as circles that represent stereotyped activities in Figure 1) that, based on the underlying meta-models, streamline the process and guarantee traceability among and between concepts.

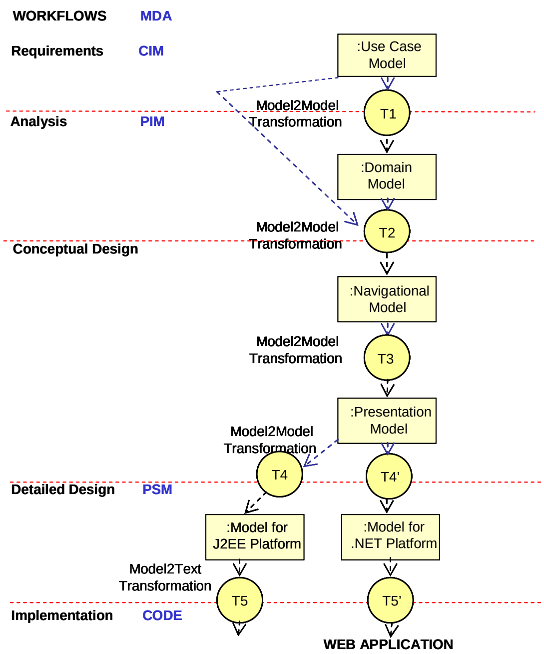

Fig. 1. Simplified version of a traditional WE Development Process.

The use of a WE process with (semi-) automatic transformations prevents some development problems such as inconsistencies among models, lack of traceability, lack of technical soundness, etc. However, this (semi-)automatic nature of the WE process also may cause the propagation of quality flaws through levels of abstraction. Hence, quality problems that nowadays are just detected at deployment time may have been introduced at any previous stage of development. As an example, the omision of certain domain relationships (which are present in the end-user's mind) in the domain model may lead to an improper navigation structure that hampers the application usability.

The six levels of refinement presented in Figure 1 imply in fact six different WE products, each one involving a different purpose of evaluation. From them, the first five can be regarded as 'internal products' in the sense that they refer to models of the product, and not the product itself, while the deployable Web application is an external product (the product that actually reaches first the testing environment and then the market). A graphical representation of the products, together with their hypothetical quality inter-relationships, is presented in Figure 2. Such relationships (still to be empirically proven by the WE community) are based on (1) the aforementioned ISO/IEC assumption that quality at one level of abstraction may be used to predict quality at lower levels of abstraction and (2) the already mentioned underlying traceability of concepts among the different WE models (see Figure 1).

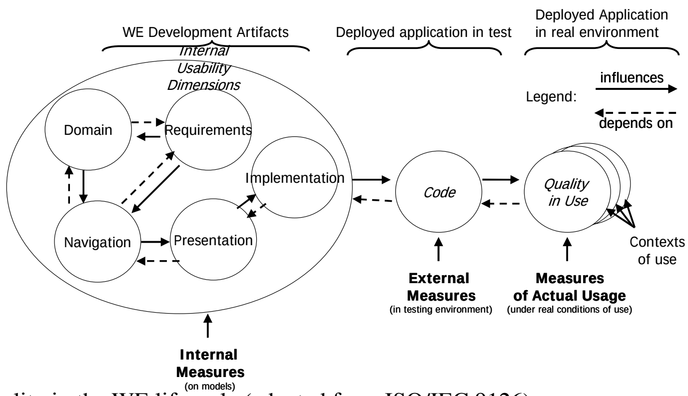

Fig. 2. Quality in the WE lifecycle (adapted from ISO/IEC 9126)

Namely, in Figure 2 we can graphically observe how the internal quality dimensions may affect an external quality dimension, that is, the quality of the final application (code) as perceived under testing conditions. Finally, such external quality may influence the actual quality of the application in real contexts of use. Next, we are presenting how we have enriched the WE process of Figure 1 to introduce quality concerns during the development of each one of the WE products.

4 Towards a Quality-Aware WE Process

Even if it is true that the ISO set of standards accompanies the definition of quality models (defined in the ISO/IEC 9126) with a software evaluation process (defined in the ISO/IEC 14598), it is a well known fact that both standards are not sufficient to direct the practitioner in the quality evaluation process. One reason for this fact may be that ISO/IEC 14598 was finished before the last version of the ISO/IEC 9126, and, while it provides generic linkages between the high-level concepts of the ISO/IEC 9126 quality models (characteristics, subcharacteristics and measures), the evaluation process is not yet specified in the format of specific prescriptive quality engineering practices. Otherwise stated, there is a mapping gap between standards and existing development process. Figure 3 presents how we propose to fill in this gap. Our proposal includes the encapsulation of each pair purpose of evaluation-product type in an independent WE quality model (dotted elements, see Figure 3) that gathers measurement concepts, attributes and measures relevant for the purpose of evaluation at such level of abstraction. Additionally, in order to preserve the semi-automatic nature of the WE process, our proposal includes the operationalization of quality models by means of a WE measurement meta-model instantiation (shadowed elements, see Figure 3). The WE measurement meta-model and the transformations that permit to calculate measurement results and, if necessary, evolve the WE models are out of the scope of this paper. Interested readers are referred to [4] for a whole description of such operationalization. The result of evolving each WE model based on a specific quality measurement model is then used at the next stage of development.

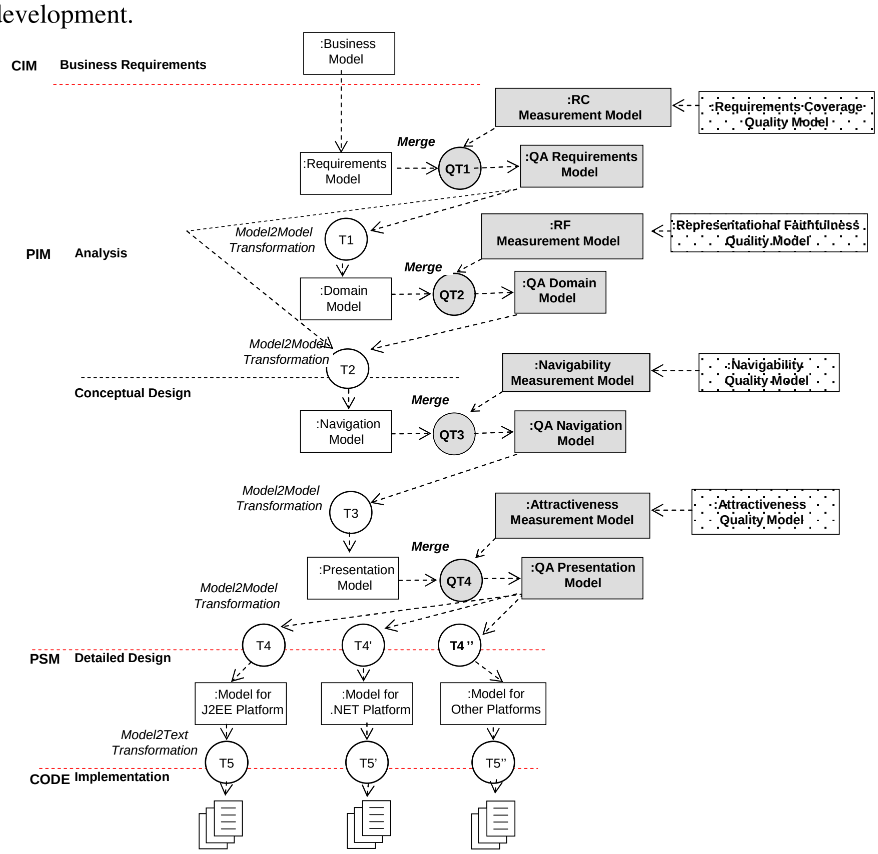

Fig. 3. Quality-aware WE Development Process

As the reader may already have inferred, the main advantages of a process such as the one included in Figure 3 are:

- It integrates quality evaluation practices in current WE practices so that each problem is detected and solved as soon as possible in the development lifecycle, what, as we have already outlined, diminishes costs and time to market of high-quality web applications.
- It is based on different WE-quality models, each one providing the most suitable basis on which to fulfill a given quality evaluation purpose. This specificity of model and evaluation purpose helps to make such models more concise. We agree with [16] in that the construction of concise quality models that are integrated in a quality management process are of prime importance to focus the quality evaluation task and carry out a comprehensive quality analysis in a very limited timeframe.
- It explicitly recognizes the relationship that, according to ISO, exists among quality of intermediate artifacts. Such relationships justify the necessity to assure that the outgoing artifact of each workflow has the required level of quality before going on to the next step of development.

As a further advantage, and in order to facilitate its adoption by industry, our proposal conforms to ISO/IEC 14598. This means that it (1) makes use of quality models and (2) it covers all the process steps defined in clause 6 of the standard. Next we are further explaining this last assertion.

4.1 Conformance to ISO 14598 set of standards

Although, in order to keep faithful to the WE philosophy, our approach is presented as an enrichment of a general WE process, it is possible to superpose the workflow defined by the ISO/IEC 14598 (much better known in industry) over the quality-aware WE process presented so far. For such superposing, the ISO/IEC 14598-3 (development) is of special interest. ISO/IEC 14598 poses two main requirements for compliance. On one hand, the quality evaluation must be based on a quality model. This end is fulfilled by our approach, as we already saw in Section 3. On the other hand, ISO/IEC 14598 demands that the evaluation process follows the following steps: (1) *Establish Evaluation Requirements* (purpose of evaluation, types of product to be evaluated and quality models), (2) *Specify the Evaluation* (select measures, establish decision criteria and establish indicators for assessment, (3) *Design Evaluation* (produce evaluation plan) and (4) *Execute Evaluation* (take measures, compare with decision criteria and assess results). Next we are presenting how we are covering each of these activities.

Establish Evaluation Requirements. The overall *purpose of evaluation* of our proposed quality-aware WE process is (using the GQM template [2]) "*analyze* the different WE artifacts *for the purpose of* assessing their conformance *with respect to* certain quality specifications *from the viewpoint of* the end-user of the application *in the context of* testing environments". This purpose is fulfilled by evaluating the main six WE models (*types of product*): requirements, domain, navigation, presentation, implementation and code. For each one of these products we need to specify a different quality model.

Specify the Evaluation. Particular applications may have specific requirements that may make necessary a tailoring of the *characteristics (measurable concepts)*, *measures, decision criteria and/or indicators* included in a given quality model. Aware of this fact, our proposal (see Figure 3) provides both general quality models defined at each level of abstraction and particular measurement models, which represent an operationalized version of the quality model. During this operationalization the designer (1) makes sure that all the relevant concepts are defined (e.g. that all measures have associated decision criteria, general or particular to the actual application being quality-evaluated) and (2) expresses the quality model in a machine-readable way (by instantiation of a well-defined WE Software Measurement Meta-model (WE-SMM)). Hence, this operationalization acknowledges the fact that certain quality elements may diverge depending on particular application domains or even particular application's quality needs. Nonetheless, the need to fulfill the restrictions imposed by such meta-model contributes to guaranteeing the correctness and completeness of the resulting instantiation. Interested readers in how such operationalization is performed are referred to [4].

Design the evaluation. In our proposal, each evaluation activity must be performed as soon as the incoming WE artifact is produced and before stepping into the next WE process activity (which makes up the *evaluation schedule*). The *evaluation method* consists on the automatic application of the WE measurement meta-model instantiation to the particular WE model we are dealing with.

Execute the evaluation. Finally, we propose to execute the evaluation in an automatic way, by means of transformation rules that interact with the WE-SMM and with the particular WE artifact meta-model to (1) get measures results, (2) calculate indicators, (3) compare indicators with decision criteria and (4) if feasible, evolve the models to improve the indicator value. Interested readers in how such execution is performed are referred to [4].

5 Conclusions and Further Work

In this paper we have proposed a WE process that integrates quality evaluation and assurance activities at every abstraction level in the development of the Web application. Building up a high-quality Web application from the end-user perspective all along the WE process implies a shift from the traditional WE quality assessment perspective to a WE Total Quality Management (TQM) approach. Briefly speaking, adopting a Total Quality Management perspective means setting the focus on preventing rather than detecting errors, with the ultimate aim of reducing the reliance on code inspections as a way of achieving quality [16]. Our assumption is that providing practitioners with WE methodologies that offer automated support for assuring quality will not only back some of the WE traditional claims of high productivity, short time-to-market and high quality, but also increase the acceptance rate of WE methodologies in industry. This in turn would provide the WE community with more data on which to refine their knowledge about when and how to use a given WE methodology.

This work is just a first step towards the inclusion of quality issues in the WE field. In order to increase the reusability of our framework, it would be necessary for the definition of generic WE-quality models to reach a consensus and identify a set of common attributes that characterize any of the WE artifacts (i.e. models) proposed by the best known WE methodologies, and center the quality evaluation on such common concepts. We do claim that such common set of concepts exist at each level of abstraction, as the recent MDWEnet initiative[1] backs. Only such attributes, together with a general definition of measures, independent from particular notations, should be included in WE-quality models in order to make them reusable among WE methodologies. Also, general tailoring rules of these quality models for certain application families should be provided in order to ease the operationalization of the quality models. Transformations should be implemented to support each of the measures included as part of those quality models, and empirical research should take place to demonstrate how the measurement results taken at early stages of development influence the quality in use of the final application. Even if we are conscious that we may never reach such a rigorous, evidence-based quality assessment, we believe that even the automatic assessment of a few empirically validated measures at each level of abstraction could already significantly increase the satisfaction of WE users.

Acknowledgments

This paper has been supported by the Spanish Ministry of Science and Technology (Grant PR2006-0374 for University Teaching Staff Stages at foreign Universities and projects CALIPSO (TIN20005-24055-E), MEC-FEDER (TIN2004-03145), ESFINGE (TIN2006-15175-C05-05), METASIGN (TIN2004-00779) and DSDM (TIN2005-25866-E)). Also, this research is part of the DADASMECA project (GV05/220), financed by the Valencia Government and the DADS (PBC-05-012-2) and the DIMENSIONS (PBC-05-012-1) projects, financed by the Regional Science and Technology Ministry of Castilla-La Mancha (Spain). Also, we would like to thank Dr. Mario Piattini for his continuous support.

[1] Interested readers can follow the lines of work and the state of evolution of this project by contacting the MDWEnet project members (http://www.pst.informatik.uni-muenchen.de/~zhangg/cgi-bin/mdwenet/wiki.cg)

References

1. Abrahao, S, Insfran, E. (2006) Early usability evaluation in Model-Driven Architecture Environments. In Proceedings of the Sixth IEEE International Conference on Quality Software, IEEE Press, Wiley, Chichester.
2. Basili, V.R., Galdiera, G. and Rombach, H.D. (1994) Goal Question Metric paradigm, in: J. J. Marciniack (Ed.), Enciclopaedia of Software Engineering, Vol 1, John Wiley and Sons.
3. Briand LL, Morasca S. and Basili V (1999) Defining and Validating Measures for Object-based High-level Design. IEEE Transactions on Software Engineering 25(5), 722-743.
4. Cachero, C.; Meliá, M.; Genero, G., Poels, G. and Calero, C. (2006). Towards improving the navigability of Web applications: a Model Driven Approach. Working Paper D/2006/7012/64. Available On-line at: www.feb.ugent.be/Fac/Research/WP/Papers/wp_06_419.pdf
5. Cachero, C., Poels, G. and Calero, C. (2007) Towards a Quality-Aware Engineering Process for the development of Web Applications. Working paper. Available On-line at http://www.dlsi.ua.es/~ccachero/pPublicaciones.htm
6. Calero, C., Ruiz, J. and Piatinni, M. (2005) Classifying web metrics using the web quality model. In Online Information Review Journal, OIR – 29,3, Emerald Literari, United Kingdom.
7. Comai S.; Matera, M.; Maurino, A. (2003) A Model and an XSL Framework for Analyzing the Quality of WebML Conceptual Schemas. In Proceedings of the International Workshop on Conceptual Modeling Quality, p 339-350. Springer-Verlag, Heidenberg, LNCS 2784
8. Garvin, D. (1984) What Does “Product Quality” Really Mean?, Sloan Management Review, Fall 1984, pp. 25-45.
9. Heuser, L (2004) The real world or Web Engineering? In Proceedings of the Fourth International Conference on Web Engineering, p 1-5, Springer-Verlag, Heidelberg, LNCS 3140.
10. ISO/IEC 9126 (2001) Software engineering – Product quality – Part 1: Quality model. International Organization for Standardization, Geneva.
11. ISO/IEC 14598 (1999). Information technology – Software Product Evaluation. International Organization for Standardization. Geneva.
12. Ivory MY (2004) Automated Web Site Evaluation. Kluwer Academic Publishers, Norwell.
13. Krogstie J., Lindland O.I. and Sindre G. (1995) Defining quality aspects for conceptual models. ISCO 1995: 216-231
14. Lang, M.; Fitzgerald, B. (2005) Hypermedia Systems Development Practices: A Survey. IEEE Software 22(2), 68-75.
15. Lindland O.I., Sindre G. and Solvberg A. (1994) Understanding Quality in Conceptual Modeling. IEEE Software Vol 11(2) pp 42 – 49. IEEE Computer Society Press. Los Alamitos, CA, USA
16. Moody D.L. and Shanks G.G. (2003). Improving the quality of data models: empirical validation of a quality management framework. Information Systems 28: 619-650. Elsevier Science Ltd.
17. Moraga, M., Calero, C., Piattini, M. (2006) Ontology Driven Definition of a Usability Model for Second Generation Portals. In Proceedings of the 1st International Workshop on Methods, Architectures & Technologies for e-Service Engineering (MATeS 2006) ACM Press. Vol. 155. ISBN:1-59593-435-9
18. Nielsen J (2000). Designing Web Usability: The Practice of Simplicity. New Riders, Berkeley.
19. Olsina, L., Rossi, G, (2002) Measuring Web Application Quality with WebQEM, In IEEE Multimedia Magazine, Vol. 9, Nº 4, pp. 20-29