Quality Issues on Model-Driven Web Engineering Methodologies

F. J. Domínguez-Mayo, M.J. Escalona, and M. Mejías

Abstract Nowadays, there are several development methodologies in the field of model-driven web engineering (MDWE) which involve different levels of model-driven architecture (MDA): CIM, PIM, PSM, or code. Attending to the high number of available methodologies, development teams may feel lost when choosing the most suitable one for their projects. Furthermore, proposals usually appear and people feel necessary to evaluate their quality in order to select the appropriate methodology or even to find out the way to improve them. This chapter presents the current work carried out in this field and it is oriented toward the definition of a framework which enables an objective measurement of the proposals' benefits.

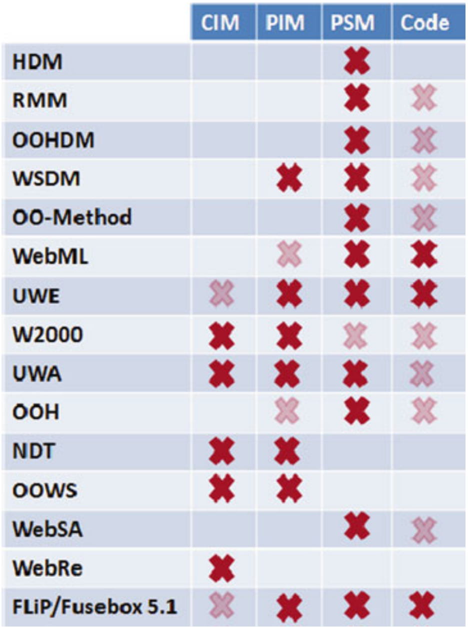
Keywords Web engineering · Quality in web engineering · Software metrics · Model-driven development

1 Introduction

Few years ago, several research groups began to analyze the characteristics of new types of software systems that emerged and were known as hypermedia systems, which have evolved to be called web systems. It was the birth of a new line of engineering software that is now called web engineering [8].

Within the paradigm of MDE (model-driven engineering), web engineering is a specific domain in which model-driven software development can be successfully applied [11]. The use of MDE in web engineering is called model-driven web engineering (MDWE), and, as can be noticed through different papers, in the last years several research groups have proposed methodologies with processes, models, and techniques to build applications [9, 14, 23, 26] and it is offering very good results [5, 10, 18, 24].

Fig. 1 Levels cover by each approach [9]



	CIM	PIM	PSM	Code
HDM			✖	
RMM			✖	✗
OOHDM			✖	✗
WSDM		✖	✖	✗
OO-Method			✖	✗
WebML		✗	✖	✖
UWE	✗	✖	✖	✖
W2000	✖	✖	✗	✗
UWA	✖	✖	✖	✗
OOH		✗	✖	✗
NDT	✖	✖		
OOWS	✖	✖		
WebSA			✖	✗
WebRe	✖			
FLiP/Fusebox 5.1	✗	✖	✖	✖

There are currently several proposals in the literature on MDWE that are very useful for building such applications.

Some of them cover most of the levels and even have tools that support the automation of transformations in the processes of development and evaluation. There are different and varied proposals in web engineering, as shown in Fig. 1 obtained from [9]. In Fig. 1 proposals are shown in rows and levels or phases in columns. A cell without a cross indicates that this approach does not consider this level in its life cycle. A shadowed red cross points out that the proposal includes a phase based on classic proposals, but does not include special proposals for the web. A dark red cross means that the proposal covers the whole level, including specific method and models for the web environment.

This diversity of possibilities and the new trend to use MDE in the proposals open a too wide range of offers and in many cases it may be complex to determine the most appropriate one. Consequently, this chapter presents a first approach to a framework which objectively assesses the proposals for MDWE that a computer has to develop and offers a choice criterion for it.

The chapter is organized into the following sections: Section 2 presents a short introduction about MDWE. Section 3 introduces the problem, motivation, and goals and tries to define a framework that permits quality evaluation of the different methodological proposals. Section 4 provides with the elements to consider in the evaluation of the approaches. Section 5, where the elements have already been identified in the previous section, offers the guidelines to both structure the assessment and focus on the work plan. Section 6 shows the methodology and work plan determined and a description of the necessary process to achieve the framework. Finally, a set of conclusions and possible future work are established.

2 Model-Driven Web Engineering

Model-driven engineering (MDE) is a software development methodology which consists in 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 [19, 20]. MDA was created with the idea of separating the specification of the operational logic of a system from the details that define its uses of the capabilities of the technological platform where it is implemented [19, 20].

Attending to the above mentioned, the goals of MDA are portability, interoperability, and reusability through architectural separation. The concept of platform independence appears frequently in MDA. Models may have the quality of being independent from the characteristics of any technological platform [27].

By applying this paradigm, the life cycle of a software system is completely covered, from requirements capture to its own maintenance, through the generation of the code. MDA distinguishes at least the following stages or levels: CIM, PIM, PSM, and code. This research focuses only on the early stages of development within the CIM and PIM MDWE field. In Fig. 2 a possible model-driven process is applied to web engineering. On the left, the MDE processes are described and on the right models in every level are showed. Orange circles in models represent transformations.

3 Problem, Motivation, and Goals

There are many proposals in the area of MDWE and many comparative studies [22]. Faced with this situation, there is a gap in decision making when an application of a methodology for a real project is required. An important need to assess the quality of existing methodologies therefore arises. On the other hand, the fact of being able to measure these methodologies may facilitate the assessment. The solution of this problem may answer the questions raised above, not only to understand the worth of a proposal but also to have an objective criterion to improve or a possibility of unifying criteria to design new proposals in the future.

The main goal of this research is to lay the basis for defining a framework that allows the quality assessment of different methodological proposals. This work points out the measurement of quality of the proposed MDWE for the first levels

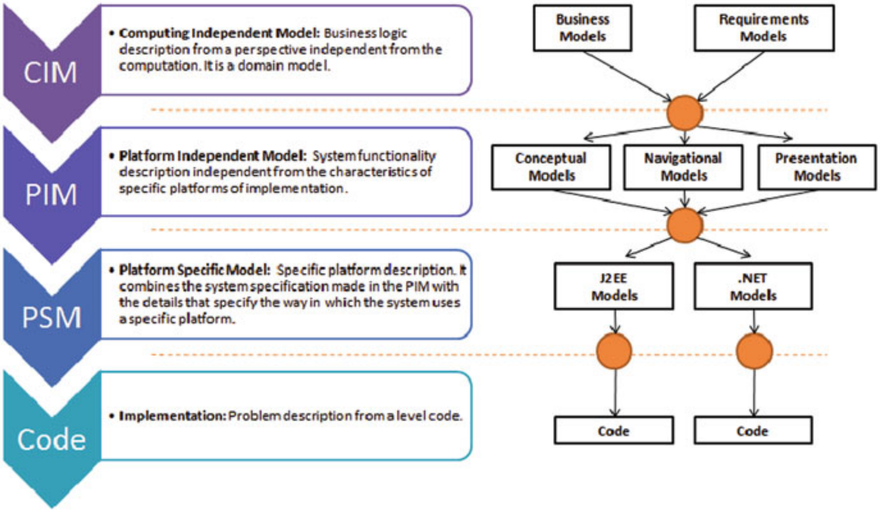

CIM
• Computing Independent Model: Business logic description from a perspective independent from the computation. It is a domain model.
PIM
• Platform Independent Model: System functionality description independent from the characteristics of specific platforms of implementation.
PSM
• Platform Specific Model: Specific platform description. It combines the system specification made in the PIM with the details that specify the way in which the system uses a specific platform.
Code
• Implementation: Problem description from a level code.
Business Models
Requirements Models
Conceptual Models
Navigational Models
Presentation Models
J2EE Models
.NET Models
Code
Code

Fig. 2 Model-driven process on web engineering

of the CIM and PIM developments, for the vast majority of existing proposals concentrate on these levels. Thus, our work concentrates on evaluating and comparing existing proposals.

The assessment is based on quantitative or qualitative values which ensure the quality of proposals. As a result, a future objective could be either to unify the criteria to decide on the use of a particular proposal in MDWE or to improve the design of new proposals and the use of standards.

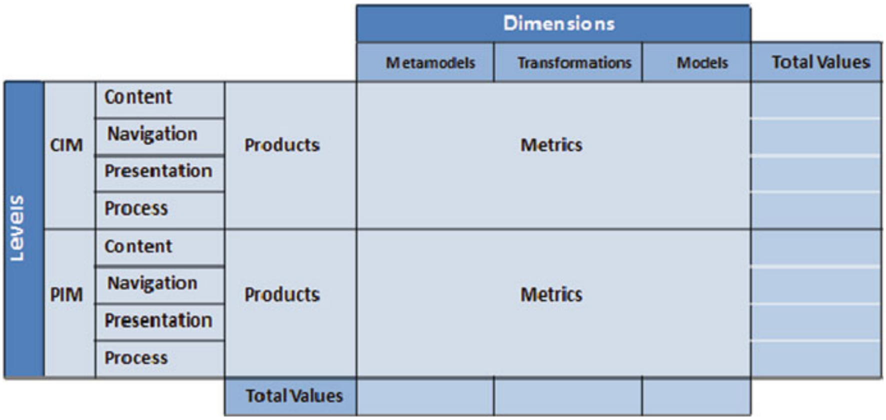
4 Factors to Consider in the Evaluation

It should be highlighted that this research work is in the early stages of development and we still have a very general description of it.

Nevertheless, it will be discussed and identified in general terms the idea of rows and columns kept in mind and considered in future works to define a table or environment necessary to solve the problem defined above. Therefore, for the environment, it would be possible to have a matrix based on the same idea as put forward in [22] which proposes a framework that permits the domain of a distributed information system to be characterized. A matrix has a series of dimensions in its columns and activities from this type of information system in its rows.

The solution of the problem is related to this same idea. The columns, rows, and cells that could become the matrix proposed are described below.

As is visualized in Fig. 3, the columns contain, however, the fundamental aspects of MDWE: metamodels, models (instances of metamodels), and transformations.



				Dimensions			
				Metamodels	Transformations	Models	Total Values
Levels	CIM	Content	Products	Metrics			
		Navigation					
		Presentation					
		Process					
	PIM	Content	Products	Metrics			
		Navigation					
		Presentation					
		Process					
			Total Values				

Fig. 3 General idea of the reference environment

For each row every product obtained from the different activities at each level may be analyzed. As it has been mentioned before, the research work will be focused on the study of the CIM and PIM levels. Regarding the study of the CIM level, there are some works which analyze each of the techniques of each activity in detail and they even perform comparative studies among different proposals. At this level, three activities are usually performed: requirements elicitation, requirements specification, and requirements validation [10].

- *Requirements elicitation*: it is the beginning of the process and the stage where all developers gather all necessary information from users and customers. There may be several sources and, as a consequence, products depending on the technique or techniques used are obtained. In [10] the technical highlights are set out: interviews, JAD (joint application development), brainstorming, concept mapping, sketching and storyboarding, use case modeling, and terminology questionnaire and comparison checklist.
- *Requirements specification:* this is the stage where requirements are defined. In the same way as in the earlier stage, products obtained depend on the specific technique. We can see in [10] the techniques used: natural language and ontology glossary, templates, scenarios, use case modeling, formal description, and prototypes.
- *Requirements validation*: at this stage users and customers validate the requirements previously specified. We see again in [10] the techniques used: walkthrough or review, audits, traceability matrix for validation, and prototyping.

Products obtained in the CIM level: the use of each of the above techniques dictates whether there are a set of products at the end of each activity, depending on the definition of each proposal and their orientation [10] (process oriented, technology oriented, or product oriented). At this particular level, NDT is a proposal that stands out since it specifies in detail the technique and product obtained in the requirements. In order to be able to measure, proposals which define products are needed and/or techniques which provide with results. In [10] a detailed study of the results obtained in different proposals is carried out. The products in the CIM level might be classified as content, navigation, presentation, or process as is shown in Fig. 3.

For the study of each PIM level, each proposal (if it is determined in this level) can define a series of products which, in the same way as in CIM, might be generally classified in content, navigation, presentation, or process, as suggested by UWE [14]. The way to classify products keeps on being defined. The products in the PIM level are the result of the application of a transformation from the CIM products.

Products obtained in the PIM level: at this level, it is also possible to classify products as content, navigation, presentation, or process like in CIM. For example, for this level, in UWE there is a metamodel for content and another for navigation, presentation, and for the business process. On the other hand, issues that are widely developed in studies such as those in [24] should be taken into account, and other factors such as the proposal maturity, web modeling, and tools should also be borne in mind. It is relevant to assess all these factors, as they may influence the decision of using a specific proposal.

Finally, the cells should have metrics that indicate either the impact or the influence of each dimension (metamodels, models, and transformations) in the product quality or performance. For example, for metamodels (being interrelated concepts) metrics which measure complexity may be considered. In regards to metrics model, an important study has been revealed in [1]. It proposes a set of metrics for navigational models to analyze the web applications' quality in terms of size and structural complexity. In this chapter, these metrics are defined and validated using a formal framework (DISTANCE) [21] for software measure construction that satisfies the measurement needs of empirical software engineering research. This framework uses the concepts of similarity and dissimilarity among software entities. In DISTANCE, the software attributes are modeled as distances (i.e., conceptual distance) between the software entities that they represent and other ones which act as reference points. These distances are then measured by mathematical functions that satisfy the axiom set of the metric space. DISTANCE could be used to define and theoretically validate all metrics (in metamodel, model, and transformation) in the framework. On the other hand, a general idea about context suitability for CIM and PIM levels might be given in total values when an approach is measured.

5 Structure of the Evaluation

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 (defined in the ISO/IEC 9126). 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

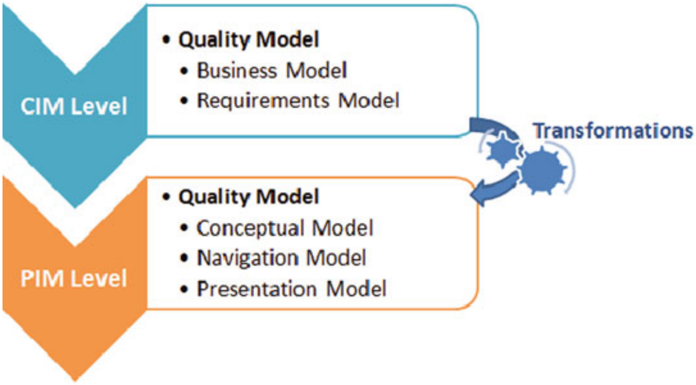


Fig. 4 Quality models which should be kept in mind in every level

and evaluating quality [3]. A different quality model must be designated for each of their products in CIM and PIM levels. In Fig. 4 different models in every level are shown for every product. It may be necessary to define a model quality for every one of them.

As far as the definition of metrics is concerned, in [4] a metrics metamodel is proposed (Fig. 5). A measure may be defined as one base or derived measure; even it may be an indicator which satisfies an information need. On the other hand, a measure is measured using a scale expressed in a unit of measurement. A measure may be defined by an attribute or a lot of them.

Furthermore, the goal/question/metric (GQM) paradigm [2] could be followed for a formal definition of these metrics. In this paradigm, a template is needed to define metrics:

- Analyze – ?
- For the purpose of – ?
- With respect to their – ?
- From the point of view of the – ?
- In the context of – ?

Our aim is to look for series of qualitative and quantitative metrics based on their nature, although it might be interesting to have standard metrics on MDWE which are all, somehow, centralized. In current literature there are many references about metrics [6, 7, 12, 13, 15–17, 25] but till now, nothing has been found to standardize all these.

Another instrument is a quality evaluation process that prescribes how and when quality evaluation must be performed [3]. For the definition of the evaluation process, Fig. 6 shows an assessment process for web engineering adapted from *ISO 14598–1*. As shown in Fig. 6, first, the evaluation requirements are established and, in steps 2 and 3, the evaluation is specified and designed, respectively. Finally, in step 4, the evaluation is executed.

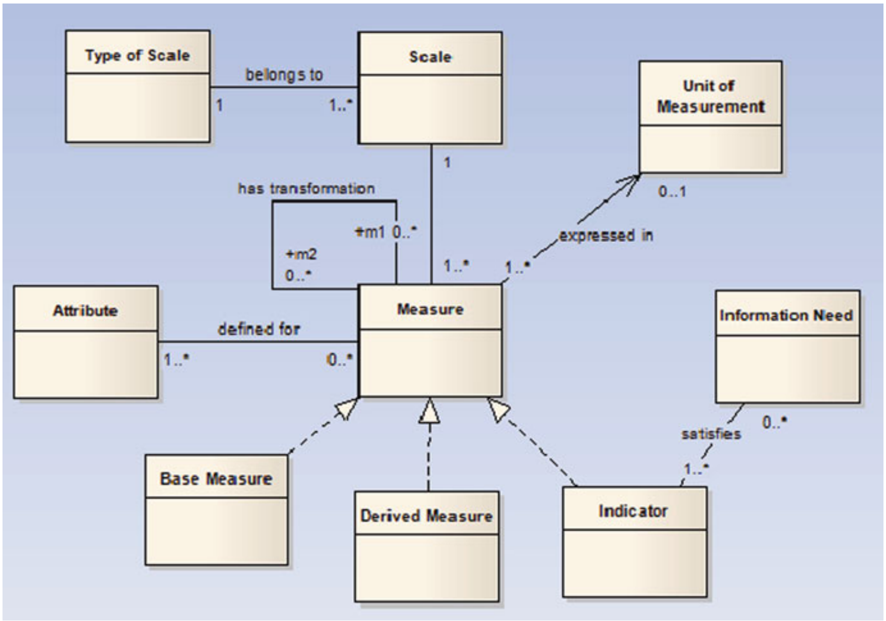


Fig. 5 Metrics metamodel

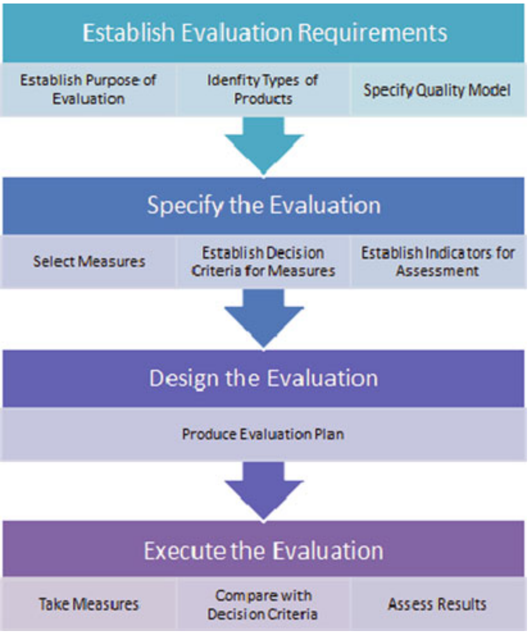

Establish Evaluation Requirements
Establish Purpose of Evaluation
Idenfity Types of Products
Specify Quality Model
Specify the Evaluation
Select Measures
Establish Decision Criteria for Measures
Establish Indicators for Assessment
Design the Evaluation
Produce Evaluation Plan
Execute the Evaluation
Take Measures
Compare with Decision Criteria
Assess Results

Fig. 6 Evaluation process adapted from ISO 14598 [3]

A controlled evaluation experiment related to this process and existing proposals could be carried out to empirically validate the suggested metrics in the framework.

Again, (GQM) paradigm could be used both to establish evaluation requirements and to specify the evaluation. On the other hand, Spearman correlation coefficient (where the variables can be expressed using an ordinal scale type) may verify the dependency among variables (metrics proposed to measure products and results like quality characteristics such as usability, accessibility, maintainability)

On the other hand, for the evaluation of the proposals, the scheme revealed in [3] which introduces a complete process (CIM, PIM, PSM, and code) to measure quality in web engineering must be applied.

6 Framework Definition Process and Conclusions

A scientific and technical research methodology is followed. The work plan includes conducting a state-of-the-art study on current topics of research:

- Metamodels, models (for instance the meta), and transformations
- Quality in metamodels, models, and transformations
- Consideration of project proposals MDWE, tools, etc.
- Comparative studies of proposals MDWE
- Frameworks designed to measure quality
- Quality in software engineering (development, models, meta, change, etc.)
- Study and definition of metrics and indicators for meta models and their transformations and metric engineering ontologies on web
- Processes for assessing quality in web engineering

The outline of the work plan corresponds to that shown in Fig. 7. First, an environment to measure the value of proposals and the definition of a process to evaluate CIM and PIM levels should be specified. In steps 2 and 3 MDWE proposals should be compared in an iterative way to obtain conclusions on the evaluations. An iterative feedback process (proposals – measurement and evaluation – conclusions) should improve as much as possible the work environment.

Bearing in mind that there are major work to be carried out and that this research is still in the early stages of development, we trust that good results will permit future research to improve the value of existing proposals in MDWE. Furthermore, the use of standards seems to be essential for the research development in this type of problem.

A framework which permits us to measure quality or adaptability for an approach or methodology given a context might be useful because it can help development teams to choose the most suitable one for every project. They would have an environment to decide which proposal is the most appropriated.

To define metrics, (GQM) paradigm might be followed for a formal definition, and these metrics might be defined and validated using the formal framework

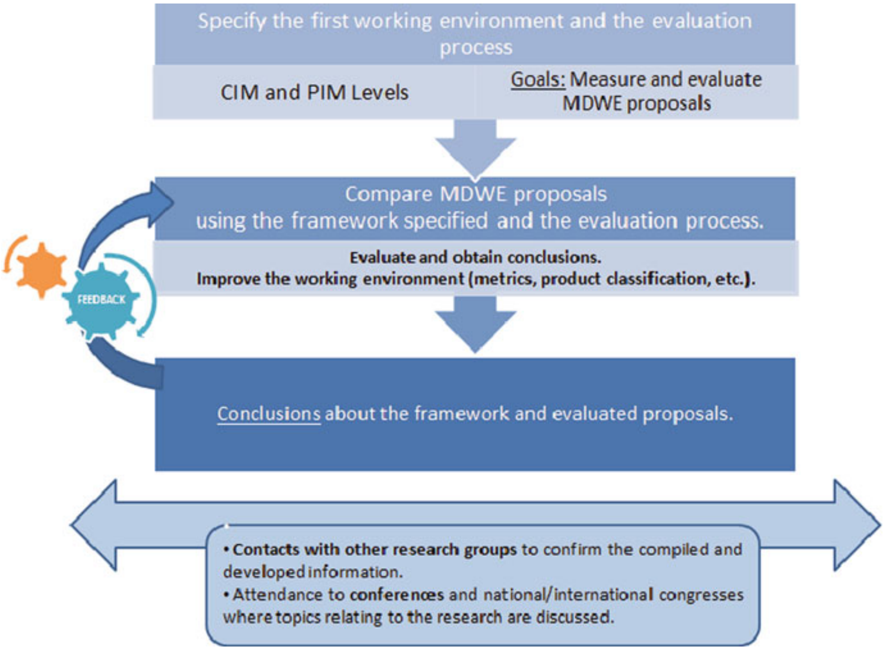

Specify the first working environment and the evaluation process
CIM and PIM Levels
Goals: Measure and evaluate MDWE proposals
Compare MDWE proposals using the framework specified and the evaluation process.
Evaluate and obtain conclusions.
Improve the working environment (metrics, product classification, etc.).
FEEDBACK
Conclusions about the framework and evaluated proposals.
• Contacts with other research groups to confirm the compiled and developed information.
• Attendance to conferences and national/international congresses where topics relating to the research are discussed.

Fig. 7 General work plan

(DISTANCE) [21]. On the other hand, quality models (for products in CIM and PIM levels in every dimension) and an evaluation model (for the complete assessment) should be defined to ensure good results.

In regards to the contributions obtained from this research, a generic environment is required for the measurement of the value of MDWE proposal in order to be able to assess and improve their quality or adaptability. In this way, criteria can be unified when developing a new methodology or improving current proposals.

Acknowledgments 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

1. S. Abrahão, N. Condori-Fernández, L. Olsina and O. Pastor (2003) Defining and Validating Metrics for Navigational Models. IEEE Computer Society. *Proceedings of the Ninth International Software Metrics Symposium (METRICS'03)*. pp. 200–210, ISSN: 1530-1435, ISBN: 0-7695-1987-3.
2. V. Basili and H. Rombach (1988) The TAME Project: towards improvement-oriented software environments. *IEEE Transactions on Software Engineering* 14, pp. 758–773.
3. C. Cachero, G. Poels and C. Calero (2007) Towards a quality-aware web engineering process. *Twelfth International Workshop on Exploring Modelling Methods in Systems Analysis and Design*, 1, pp. 7–16. Held in conjunction with CAISE'07 Trondheim.

4. C. Cachero, G. Poels, C. Calero and Y. Marhuenda (2007) Towards a quality-aware engineering process for the development of web applications. Working Papers of Faculty of Economics and Business Administration, Ghent University, Belgium 07/462, Ghent University, Faculty of Economics and Business Administration.
5. C. Calero, J. Ruiz and M. Piattini (2004) *A Web Metrics Survey Using WQM*. ICWE 2004, LNCS 3140, pp. 147–160.
6. S.R. Chidamber and C.F. Kemerer (1991) Towards a metrics suite for object oriented design, in A. Paepcke, (ed.) *Proc. Conference on Object-Oriented Programming: Systems, Languages and Applications (OOPSLA 91)*. ACM, New York, NY, USA. Vol. 26, Issue 11, pp. 197–211, ISSN:0362-1340.
7. S.R. Chidamber and C.F. Kemerer. (1994) *A Metrics Suite for Object Oriented Design, IEEE Transactions on Software Engineering*. IEEE Press Piscataway, NJ, USA, Vol. 20, Issue 6, pp. 476–493, ISSN:0098-5589.
8. Y. Deshpande, S. Marugesan, A. Ginige, S. Hanse, D. Schawabe, M. Gaedke and B. White (2002) Web Engineering, *Journal of Web Engineering*, 1(1), pp. 3–17.
9. M.J. Escalona and G. Aragón (2008) *NDT. A model-driven approach for web requirements*. IEEE Transactions on Software Engineering, San Francisco, CA, USA, pp. 377–390, ISSN: 0098-5589.
10. M.J. Escalona and N. Koch. (2004) Requirements engineering for web applications – a comparative study. *Journal of Web Engineering*. 2(3), pp. 193–212.
11. J. Fons, V. Pelechano, M. Albert and O. Pastor (2003) Development of web applications from web enhanced conceptual schemas, *Proceedings of the 22nd International Conference on Conceptual Modeling*. I.-Y. Song et al. (Eds.): ER 2003, LNCS 2813, pp. 232–245.
12. F. García, M. F. Bertoa, C. Calero, A. Vallecillo, F. Ruíz, M. Piattini and M. Genero (2005) Towards a consistent terminology for software measurement. *Information and Software Technology*. 48, pp. 631–644.
13. B. Henderson-Sellers (1996) *Software Metrics*, Prentice Hall, Hemel Hempstaed, UK.
14. C. Kroiβ and 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.
15. A. Lake and C. Cook (1994) Use of factor analysis to develop OOP software complexity metrics. *Proceedings of The 6th Annual Oregon Workshop on Software Metrics*, Silver Falls, Oregon.
16. Y.-S. Lee, B.-S. Liang, S.-F. Wu and F.-J. Wang (1995) Measuring the coupling and cohesion of an object-oriented program based on information flow, *Proc. International Conference on Software Quality*, Maribor, Slovenia.
17. M. Lorenz and J. Kidd (1994) *Object-Oriented Software Metrics*, Prentice Hall Object-Oriented Series, Englewood Cliffs, NJ.
18. N. Moreno, P. Fraternalli and A. Vallecillo (2006) *A UML 2.0 Profile for WebML Modeling*, ICWE'06 Workshops.
19. OMG: MDA Guide (2005) http://www.omg.org/docs/omg/03-06-01.pdf
20. J. M. Pérez, F. Ruiz, M. Piattini. (2007) *Model Driven Engineering Aplicado aBusiness Process Management*, Informe Técnico UCLM-TSI-002.
21. G. Poels and G. Dedene. (1999) *DISTANCE: A Framework for Software Measure Construction*. Research Report 9937, Department of Applied Economics, Catholic University of Leuven
22. J. Ralyté, X. Lamielle, N. Arni-Bloch and M. Lèonard (2008) Distributed Information Systems development: A Framework for Understanding and Managing. *International Journal of Computer Science and Applications, Technomathematics Research Foundation,* 5(3b), pp. 1–24.
23. A. Schauerhuber, M. Wimmer and E. Kapsammer (2006) Bridging existing web modelling languages to model-driven engineering: a metamodel for WebML. *International Conference On Web Engineering; Vol. 155. Workshop proceedings of the sixth international conference on Web engineering*. Palo Alto, CA (MDWE'06). ISBN:1-59593-435-9.

24. 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*, 4(3), pp. 234–305.
25. Sdmetrics, http://www.sdmetrics.com/
26. A. Vallecillo, N. Koch, C. Cachero, S. Comai, P. Fraternali, I. Garrigós, J. Gómez, G. Kappel, A. Knapp, M. Matera, S. Meliá, N. Moreno, B. Pröll, T. Reiter, W. Retschitzegger, J. E. Rivera1, A. Schauerhuber, W. Schwinger, M. Wimmer and G. Zhang (2007) MDWEnet: A Practical Approach to Achieving Interoperability of Model-Driven Web Engineering Methods. *:"7th International Conference on Web Engineering, Workshop Proceedings"*, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy, pp. 246–254, ISBN: 978-88-902405-2-2.
27. Wikipedia, http://en.wikipedia.org/wiki/Model-driven_engineering (May 2009).