Model-Driven Web Engineering

Nora Koch, Santiago Meliá-Beigbeder, Nathalie Moreno-Vergara, Vicente Pelechano-Ferragud, Fernando Sánchez-Figueroa, and Juan-Manuel Vara-Mesa

The continuing emergence of new platforms and technologies make Web applications resemble more and more complex desktop applications. Therefore, the tasks of designing, implementing and adapting the Web applications are becoming a more time-consuming and error-prone issue. In the Web Engineering context, Model-Driven Development (MDD) and Model-Driven Architecture (MDA) principles are being used to successfully address the construction, evolution and adaptation of Web applications. In this article we show how the Model-Driven Web Engineering (MDWE) discipline has arisen and how MDD/MDA principles are applied in the Web Domain to define models and metamodels, to specify model transformations, to manage interoperability issues and to build tools that support the development process.

Keywords: CASE Tools, Interoperability, Metamodels, Model Driven Architecture (MDA), Model Driven Development (MDD), Model Transformations, Web Engineering.

1 Introduction

Since 2001 the Model-Driven Architecture (MDA) initiative has been applied to many application domains showing that, in general, it works better in those domains dominated by functional requirements, with well structured models, precise separation of concerns and standard platforms. Web Engineering, specifically, has proved to be an application domain where MDA has brought significant benefits. In particular, MDA allows successfully addressing interoperability, model evolution and adaptation problems of Web systems, as emerging new platforms and changes in technologies constantly happen in this area.

In this sense, we have seen how many Web Engineering approaches are shifting to becoming "*MDA compliant*", which has resulted in some major changes in their notations, processes and tools. In particular, some of these approaches have had to: (a) redesign their Web modelling languages using meta-modelling techniques, rather focusing on notational aspects of the languages being used; (b) reorganize their original set of models in a modular and platform independent manner; (c) reformulate their development processes in terms of model transformations and model merges; and (d) incorporate and adopt standards that support the realization of the MDA initiative such as UML® (Unified Modelling Language), MOF (Meta-Object Facility), XMI (XML Metadata Interchange), or QVT (Query/View/Transformations)[1] .

However, despite all these challenges, the benefits of the adoption of Model-Driven Development (MDD) and MDA ideas and techniques, to Web Engineering has far outweighed its costs. MDA has provided the opportunity to inject good software engineering practices into the Web applications domain; and has allowed successful bridging

[1] OMG. <http://www.omg.org>

Authors

Nora Koch has been a research assistant at the Ludwig-Maximilians-Universität (LMU) of Munich (Germany) since 1995 and has worked at F.A.S.T. Applied Software Technology GmbH as consultant and project manager since 1998. She received her Ph.D. in Computer Science from the LMU in 2000. Nora is leader of the Web Engineering Group at the LMU, responsible for the development of the UWE methodology. Her main research interests focus on methods for the development of Web applications, personalised Web systems and model-driven engineering. Nora organised the 4th International Conference on Web Engineering (ICWE 2004) in Munich, two editions of the Model-Driven Web Engineering Workshop (MDWE 2005 and MDWE 2007) and is a founder of the network MDWEnet. She is a member of the IEEE and the German Society of Informatics. Nora has published more than 70 publications. <kochn@pst.ifi.lmu.de>.

Santiago Meliá-Beigbeder is Assistant Professor at the Department of Languages and Information Systems at the University of Alicante (Spain), where he is the director of a postgraduate Master in Software Engineering. Also, he is a member of the IWAD research group in the same University. His research interest includes Model-Driven Development, Web Engineering Methodologies, Automatic Code Generation Techniques and Web Software Architecture, all of which were part of his Ph.D. received at the University of Alicante in 2007. He has published in prestigious journals such as the European Journal of Information Systems, Journal of Web Engineering, Revue d'Interaction Homme-Machine and Lecture Notes in Computer Science, and at conferences (OOPSLA, EC-MDA, ER, EC-Web, ICWE, CADUI, etc.). He regularly serves on the Program Committees of several international conferences and workshops. <santi@dlsi.ua.es>.

Nathalie Moreno-Vergara is Assistant Professor in the Department of Computer Science of the University of Málaga (Spain) where she received her MSc. and is completing a Ph.D. in Computer Science. Her research interest is mainly oriented towards model-driven development and, in particular, she is interested in investigating conceptual modeling methodologies, business process modeling, model transformation languages and code-generation techniques in the Web applications context. Her

of the previously existing gap between the high level design models and concepts and the low-level Web implementation code (Preciado et al [1]). This has led to a discipline within the Web Engineering called Model-Driven Web-Engineering (MDWE) that focuses, among others, on the interoperability of the currently existing methodologies for the development of Web applications. Worthy of mention is the MDWEnet[2] initiative started by a group of European researchers working on MDWE, with the objective of improving the interoperability of MDWE approaches and tools in order to widen their scope and provide better tools and methods to the industry.

The adoption of MDD/MDA by the Web Engineering community is not free from problems and limitations. In this article, we shall give a critical overview of the state of the art in MDA-based Web Engineering as currently perceived by some of the groups that actively work on it. Not only the efforts and results already achieved will be described, but also the challenges, threats and weaknesses of this approach will be identified based on our experiences and developed systems, with the aim of helping MDA to evolve towards a more mature and successful development approach for Web systems.

The paper is structured as follows. Section 2 describes the role achieved by models and meta-models in the abstraction and design process of current Web applications as a consequence of the MDA goal of automatically generating implementations from models. In this context, Section 3 illustrates the way in which model transformations are used within the development process. A brief report on tools that support MDD/MDA principles in the Web domain is given in Section 4. Section 5 shows how the Web Engineering community has addressed the difficult problems of interoperability using the MDA concepts and mechanisms. Finally, Section 6 points out current strengths, weaknesses and major challenges that could considerably improve the efficiency of using MDA in the Web context.

2 Models and Metamodels in the Web Domain

MDA is based on the construction and transformation of models which represent a computational independent viewpoint (CIM), a platform independent viewpoint (PIM) or a platform specific viewpoint (PSM). During recent years, the Web engineering community has proposed several methods for modelling Web applications that mainly focus on the construction of PIMs. Web Engineering methods, Rossi et al. [2] like Hera, OOHDM, OOWS, UWE, WebML, WSDM, and methods such as MIDAS [3], OO-H and WebSA (for both see Meliá et al [4]) propose building similar types of models but using different graphical notation for the representation of these models. Most of them use a proprietary notation; some combine the use of the Unified Modeling Language (UML) with their own notation, and only UWE and WebSA use UML 2.0 for all its models. UWE uses plain UML as far as possible and defines for the Web domain features a UML profile following the extension mechanisms provided by UML. As a result, UWE is a UML

most significant publications can be found in relevant journals (the European Journal of Information Systems, the Journal of the American Society for Information Science and Technology or the ET Software), book chapters in Springer Verlag and well-known international conferences such as ICWE, DEXA, MODELs, ICSE, etc. She usually participates as part of program committees and reviewer of conferences and workshops in her research area. <vergara@lcc.uma.es>.

Vicente Pelechado-Ferragud is Associate Professor in the Department of Information Systems and Computation (DISC) at the Valencia University of Technology (Spain). His research interests are web engineering, conceptual modelling, requirements engineering, software patterns, web services, ambient intelligence, business process modeling and model driven development. He received his Ph.D. from the Valencia University of Technology in 2001. He is currently teaching software engineering, design and implementation of web services, design patterns, and model driven development and code generation in the Valencia University of Technology. He is a member of the OO-Method Research Group at the DISC. He has published in several well-known scientific journals (Information Systems, Data & Knowledge Engineering, Information and Software Technology, Internacional Journal of Web Engineering and Technology, etc.), book chapters in Springer, IDEA, M. Sharpe and international conferences (ER, CAiSE, WWW, ICWE, EC-WEB, WISE, AH, ICSOC, DEXA, etc.). He has been a member of Scientific and Organizing Committees of well-known International Conferences and Workshops (CAiSE, ER, ICWE, ICEIS, ACM MT and IADIS). <pele@dsic.upv.es>.

Fernando Sánchez-Figueroa is a professor in the Department of Computer Science at University of Extremadura (Spain), where he teaches Concurrent Programming and Design Patterns. Sánchez has a MSc. in computer science from the University of Sevilla (Spain) and a Ph.D. in computer science from the University of Extremadura. His research focuses on Web Engineering, specifically on Rich Internet Applications, Web Acessibility and Model Driven Web Engineering. He has been Vice-rector of ICT at the University of Extremadura for four years. Currently he holds a "Software Engineering" chair sponsored by INSA-IBM. < fernando@unex.es>.

Juan-Manuel Vara-Mesa obtained his BSc. and MSc. in Computer Science Engineering at the *Universidad Rey Juan Carlos* (Madrid, Spain), where he did the Doctoral Courses on the Computer Science and Mathematical Modeling Program. Currently he works as assistant professor in the department of Informatics Languages and Systems II of the *Universidad Rey Juan Carlos* and he is a member of the Kybele Research Group where he is doing his Ph.D. thesis focused on Model-Driven Engineering for the development of Web Information Systems. He is co-author of several books and book chapters, as well as several publications at national and international events and journals, and he has participated on several regional, national and European research projects. <juanmanuel.vara@urjc.es>.

[2] MDWEnet. <http://www.pst.ifi.lmu.de/projekte/mdwenet/>.

compliant Web domain specific language. There is a clear tendency to the use of UML by other methods as well, mainly due to the advantages provided in the use of CASE tools and meta-modelling for model-driven approaches.

A main characteristic shared by all these Web engineering approaches is the *separation of concerns*. This characteristic allows building different models to address a variety of concerns relevant in the Web domain like content of the Web application, hypertext structure, presentation aspects, adaptivity (in particular personalisation and context-awareness) and Web architectural issues. Models are merged based on integration mechanisms either implemented in proprietary tools or by the application of general rules defined in model transformation languages. In addition, MDWE approaches follow the MDA principles creating CIMs such as the requirements models built by WebRE [5] and OOWS; building PIMs for navigation, presentation and business process specification (almost all methods); building PIMs for architectural models (WEI and WebSA); and obtaining PSMs or transforming into code for specific platforms such as Java EE, Struts, Spring and .NET. The general objective is that in the development process the creation of models that take into account technological aspects is postponed as long as possible. The main advantage is to be able to react efficiently and with low costs to technology changes.

The entities used in all these Web specific models, and the relationships between entities can be also represented as a model, a so-called metamodel, which is needed for the specification of the model transformations. The first metamodel in the Web domain was presented for the UWE approach at the 3rd International Conference on Web Engineering (ICWE 2003) [6]; other methods defined then their metamodel; recently Moreno and Vallecillo [7] proposed a common reference metamodel called Web Engineering Interoperability (WEI), which is described as a MOF metamodel. WEI also defines *lightweight* extensions of UML, i.e., UML profiles, for representing the specific syntax for each of its metamodels.

3 Model Transformations for Generating Web Applications

As the MDA framework suggests, the transformations could be applied to establish a traceable development process from abstract models (CIM or PIM) to the platform dependent models (PSM) or directly to the implementation. Thus, many Web engineering approaches have defined transformations to obtain some parts or the entire implementation of a Web application. As it is well known, a domain-specific strategy such as the Web domain, allows significant parts of the implementation to be generated automatically and to reduce the development effort [8]. Several Web methods have taken advantage of this aspect by developing commercial CASE tools as presented in the next section. These CASE tools use code generation techniques to obtain Web applications from a reduced set of conceptual or design models. Within the model transformations scope we can distinguish between model-to-model transformations and model-to-code transformations. Currently, most Web methods are starting to use model transformations to extend or to implement CASE tools in order to take advantage of the opportunities which transformation languages can provide.

Next, we present how the different Web methods have applied model-to-model and model-to-text transformations to produce Web applications.

3.1 Model-to-Model Transformations

There are two types of model-to-model transformations: (1) vertical transformations that convert models from higher into lower level of abstraction and (2) horizontal transformations which describe mappings between models of the same level of abstraction.

Historically, in the Web domain, most cases of vertical transformations have been developed using using tool-specific proprietary languages. More recently though, Web approaches such as UWE [9], OOWS [10], WebSA [4] and MIDAS [3] have formalized all or part of their development process using model-to-model transformations languages such as QVT, ATL or AGG. In some cases, these model-to-model transformations are defined as merge mechanisms to introduce new concepts like architectural styles (WebSA), user requirements (WebRE [5] and OOWS Requirements Analysis [10]) and quality measures [11]. In other cases, they have been established to formalize the mappings from the original process (UWE [12]).

The horizontal transformations have been applied in the Web domain to maintain the consistency of the model specifications, checking that the different models do not impose contradictory requirements on the elements they share [7]. However, due to the novelty of these models, there is a lack of maturity in their current standards and tools. Therefore, some of the properties of these transformations (reusability, easy of use and a reduced maintainability) have not been tackled yet. Furthermore, there is no Web Engineering commercial CASE tool available that is completely based on model-to-model transformations.

3.2 Model-to-Code Transformations

The Web engineering community has extensive experience in model-to-text transformations or code generation. Approaches such as WebML, OOH, OO-Method/OOWS have been generating Web applications for almost ten years now. In some cases, code generation is realized using general-purpose languages (C++, XSLT, Java or Python) which do not cover some desirable requirements for the model-to-text transformations. Recently, an OMG standard Model-to-Text Request-For-Proposal has established the proper characteristics of the model-to-text languages (e.g. Round-Trip engineering) and some proposals such as Xpand and MOFScript have been adapted to it with mixed success. Recently, OOWS and WebSA have used MOFScript and Xpand in the implementation of their code generators in order to benefit from its properties.

4 CASE Tools supporting MDWE

In this section we give a brief overview of the existing CASE tools supporting MDWE proposals and we make a short review of the main problems and challenges to be tackled in this field. First of all, it should be noticed that we make a distinction between UML based and non-UML based tools.

On the one hand, we consider those CASE tools developed to work with domain specific languages that extend the UML standard (UML profiles). The most relevant CASE tool that falls in this category is ArgoUWE [13]. Initially, ArgoUWE was developed to extend the open-source tool ArgoUML by adding features for modelling content, navigation, and presentation structures. Later on, features were added to model the business logic and the behaviour of workflow-driven Web applications and to detect inconsistencies in UWE models based on the UWE metamodel and its OCL well-formed rules [13]. Currently the main problem of this approach is that ArgoUML and therefore ArgoUWE does not support UML 2.0. WebTE [14] is a UML tool that supports XMI and allows introduction of the OO-H and WebSA models and transformations into a transformation engine which executes them and produces a Java EE Web application. Currently there exists other solutions based on the use of UML, like the AriadneTool [15], the DaVinci framework [16], MIDAS-CASE [17] or VisualWade [18], but they either offer just some limited functionalities or have been closed.

On the other hand, we consider those tools developed to work with languages for Web applications modelling and deployment defined by means of MOF-based metamodels. The tools in this group may be considered not fully "*MDA conformant*", in the sense that they are not UML-based tools. Currently the most representative of these is WebRatio [19]. WebRatio is a commercial tool developed for giving support to the WebML method that is focused on the development of data-intensive applications. With WebRatio, the business objects are modelled using the UML or E/R standards while the front-end is modelled using WebML. Then, the entire application for the Java EE architectures and SQL/XML data sources is automatically generated from those models. There are also more recent non-UML CASE tools, like M2DAT [3] or the OOWS Suite [20], but they are still under development.

When talking about CASE tool support it should be noted that the proliferation of technologies and tools for developing "*your own*" MDD tools is facilitating the adoption and implementation of MDA principles and techniques. Many software companies and research groups are considering the development of their own CASE tool for supporting their own MDWE method (following the MDA, Software Factories, Product Lines, Generative Programming of whatever other more specific model driven proposal). This way, technology is playing a key role in the distinction between UML based and non-UML based tools: the facilities provided in the context of the *Eclipse Modelling Project* (EMP) and other DSL frameworks, like the *Generic Modelling Environment* (GME) or the *DSL Tools*, have shifted the focus from UML-based approaches to MOF-based ones.

Special attention has to be paid to the EMP. The quantity and quality of the MDD facilities provided in the context of this project (a common modelling framework like EMF, meta-editors like GMF, transformation engines like ATL or VIATRA, code generators like MOFScript) has given rise to a new generation of Eclipse tools. As a consequence, more and more MDWE proposals are developing their tools as Eclipse plug-ins, like the OOWS suite and M2DAT, or at least, upgrading or re-defining them to be "Eclipse compliant", like WebRatio [21].

However, there is still a lot of work to do. A very common problem, clearly stated by Moreno and Vallecillo [2], is that the mapping rules are typically hard-coded in the CASE tool (e.g. this is the case of ArgoUWE and WebRatio). This fact results in a gap between the design of the Web application and the final implementation. According to MDA principles, these rules should be defined at a more abstract level, using the QVT standard. Although some proposals have already tackled this task (see [13] for UWE, [4] for WebSA and [7] for WEI), these improvements have still to be integrated in the corresponding CASE tools. The lack of a reference implementation for QVT (which has led to non-complete QVT parsers for a subset of this language [7]) complicates this integration. Another problem is interoperability, in this sense, the use of *weaving* models to automate model migration is becoming widely accepted. Vara et al [22] shows how to apply this approach in a real industrial environment. Such an approach is being studied as a way to automate tools interoperability. Finally, the reader should notice that even though MDD is a widely accepted approach, MDWE is still relatively new: all the tools listed in this section are academic proposals. So, we can conclude that the most outstanding challenge for the developers of MDWE CASE tools is to take their tools from academic to industrial environments.

5 Interoperability Issues in MDWE

As stated in previous sections, current model MDWE approaches provide a set of methods and supporting tools for a systematic design and development of Web applications. However, these proposals have some limitations, especially for exchanging models or representing further modelling concerns, such as architectural styles, technology independence, or distribution. A possible solution to these issues is providing interoperability among MDWE proposals, enabling them to complement each other. Interoperability is at the heart of MDA at different levels: models and metamodels, transformations and tools.

Regarding models and metamodels, there exist three possibilities for achieving this interoperability: a) obtaining a unified method based on the strengths of the different methods; b) obtaining bridges for interoperability between the individual models and tools and c) obtaining a common metamodel. All these possibilities have their own benefits and disadvantages. Currently there are two projects in progress regarding options b) and c). In [23] Wimmer et al., presented a methodology based on MDA for making interoperable three Web modelling approaches (WebML, OO-H and UWE). They used the Ecore implementation of the MOF standard for the definition of the three metamodels. ATL is used as model transformation language to implement the transformation rules and an ATL engine executes the transformation. The next step for them is defining a common metamodel for Web Modeling. Another promising approach is WEI [7], a model-based framework for building Web applications that, among other goals, tries to provide a common framework (and metamodel) in which current proposals could be integrated and formulated in terms of the MDA principles. WEI can be instantiated both to build Web applications from scratch, and to build Web applications based on existing models (including those defined using other methodologies, e.g., UWE, WebML or OO-H).

Regarding transformations, we find a problem due to each approach using its own script language that is incompatible with other languages and tools that users often use. In this sense, QVT is not being used as thought by the OMG as stated in the previous section.

Regarding tools interoperability, despite the efforts of the OMG, the XMI standard has proven to be unsuccessful, especially when working with UML profiles. Until that happens it seems more convenient to take advantage of model-to-model transformations to achieve CASE tools interoperability, whether or not they are UML based tools.

6 Future Challenges

MDWE methods are evolving to be properly adapted to the continuous evolution of Web system requirements and technology. In the last few years a new type of Web applications, known as Web 2.0 has emerged. These applications let people collaborate and share information online. Murugesan [2] says the *People-Centric Web* and the *Read/Write Web*, offers smart user interfaces and built-in facilities for users to generate and edit content presented on the Web and thereby enrich the content base.

Service Oriented applications and the Web 2.0 are providing a clear infrastructure to integrate multiple software services under a rich user interface. AJAX-based (Asynchronous Javascript and XML) Rich Client Applications or RIAs, Service Mashups, REST or XML Web Services allow integrating current Web applications with third party services, portals, and with legacy systems. RIAs are changing the browser from a static request-response interface to a dynamic, asynchronous interface. RIAs promise a richer user experience and a set of benefits that affect Web Engineering methods [24].

The wide adoption of mobile devices allows users to access the Web using handheld devices (pocket PCs, PDAs, smartphones, etc). Mobile Web applications offer some additional features compared to traditional desktop Web apps such as location-aware services, context-aware capabilities and personalization. We are not forgetting that Web applications development is a complex task that also needs to take into account many different aspects and non functional requirements such as scalability, reliability, availability, evolution and maintainability, usability, security, accessibility, mobility, localization, personalization, adaptivity, etc.

Web Usability is one of the most relevant quality factors that should be taken into account by current and future model driven Web engineering methods. Usability should be integrated in each step of the software development method in a mandatory way, because generated Web apps should meet the diverse expectations and needs of many types of users, including able and disabled users, with different skills. The continuing appearance of new technologies like RIAs and the Mobile Devices does not guarantee that current design guidelines and usability tests will work for those new approaches to assure a better user experience.

However, MDA is still a relatively young approach and there are some issues around the MDA-based software development not yet sufficiently clarified. In particular, MDA is originally intended for new developments and it is not clear how MDA handles the production of releases, patches or updates, an important part of the ongoing maintenance of Web applications. Regarding non-functional requirements, MDA deals well with functional properties but it is less capable of dealing with non-functional requirements. In this sense, there is a debate about whether the specification of non-functional requirements is within the scope of MDA. In summary, while the principles underlying MDA are sound, there remain issues that must be solved for MDA to realize its full potential, particularly in the Web applications domain.

References

[1] H.W. Gellersen, M. Gaedke. Object-Oriented Web Application Development. IEEE Internet Computing, 1999. Vol 3(1), pp. 60-68.

[2] G. Rossi, O. Pastor, D. Schwabe, L. Olsina (eds.). Web Engineering: Modelling and Implementing Web Applications. Springer Human-Computer Interaction Series, 2007. ISBN: 978-1-84628-922-4.

[3] P. Cáceres, V. De Castro, J.M. Vara, E. Marcos. Model Transformations for Hypertext Modelling on Web Information Systems. Proc. of the ACM/SAC 2006 Track on Model Transformations (MT2006), Dijon, France, 2006, pp. 1256–1261.

[4] S. Meliá, J. Gómez. The WebSA Approach: Applying Model-Driven Engineering To Web Applications. Journal of Web Engineering (JWE), 5(2), 2006, pp. 121–149.

[5] M.J. Escalona, N. Koch. Metamodelling the Requirements of Web Systems. Web Information Systems and Technologies: Int. Conferences WEBIST 2005 and WEBIST 2006. Revised Selected Papers, Springer LNBIP, Vol. 1, 2007, pp. 267-280.

[6] N. Koch, A. Kraus. Towards a Common Metamodel for the Development of Web Applications. Proc. 3rd Int. Conf. Web Engineering (ICWE 2003), LNCS 2722,

497-506. Springer Verlag, 2003.

[7] N. Moreno, A. Vallecillo. Towards Interoperable Web Engineering Methods. Accepted for publication at the Journal of the American Society for Information Science and Technology (JASIST), 2008.

[8] J. Bettin. Measuring the Potential of Domain-Specific Modeling Techniques. Proc. of the Second Domain-Specific Modeling Languages Workshop, OOPSLA, Working Papers W-334. Helsinki School of Economics, 2002 , pp. 39-44.

[9] N. Koch. Transformation Techniques in the Model-Driven Development Process of UWE. Proc. of the 6th Int. Conf. on Web Engineering (ICWE 2006), ACM Vol. 155, Palo Alto, California, 2006.

[10] P. Valderas, V. Pelechano, O. Pastor. A Transformational Approach to Produce Web Application Prototypes from a Web Requirements Model. International Journal of Web Engineering and Technology (IJWET), Vol. 3, No. 1, 2007, pp. 4-42.

[11] C. Cachero, S. Melia, M. Genero, G. Poels, C. Calero. Towards Improving the Navigability of Web Applications: A Model-Driven Approach. European Journal of Information Systems, 2007, Vol. 16, pp. 420-447.

[12] A. Kraus. Model Driven Software Engineering for Web Applications. PhD Thesis, Institut für Informatik, Ludwig-Maximilians-Universität München, 2007.

[13] A. Knapp, N. Koch, G. Zhang, H.-M. Hassler. Modeling Business Processes in Web Applications with ArgoUWE. Proc of Int. Conf. Unified Modeling Language (UML 2004), Springer LNCS 3273, 2004, pp. 69-83.

[14] S. Meliá, J. Gómez, J.L. Serrano. WebTE: MDA Transformation Engine for Web Applications. Proc. 7th Int. Conf. Web Engineering (ICWE 2007), Springer LCNS 4607, Como, Italy, 2007.

[15] P. Diaz, S. Montero, I. Aedo. Modelling Hypermedia and Web Applications: the Ariadne Development Method. Information Systems, Vol.30(8), 2005, pp. 649-673.

[16] A. Langegger, J. Palkoska, R. Wagner. DaVinci - A Model-driven Web Engineering Framework. J. Web. Infor. Syst Vol.2(2), 2006 pp. 119-132.

[17] J.M. Vara, V. De Castro, E. Marcos. WSDL Automatic Generation from UML Models in a MDA Framework. International Journal of Web Services Practices, Vol.1, No.1-2, 2005, pp. 1-12.

[18] J. Gómez, A. Bia, A. Parraga. Tool Support for Model-Driven Development of Web Applications. Web Information Systems Engineering (WISE 2005), pp. 721-730.

[19] Webratio. <http://www.webratio.com>.

[20] J. Fons, V. Pelechano, O. Pastor, P. Valderas, V. Torres. Applying the OOWS Model-Driven Approach for Developing Web Applications. The Internet Movie Database Case Study.

[21] R. Acerbis, A. Bongio, M. Brambilla, S. Butti. WebRatio 5: An Eclipse-Based CASE Tool for Engineering Web Applications. Web Engineering, 2007, pp. 501-505.

[22] J.M. Vara, M. Didonet Del Fabro, F. Joualt, J. Bezivin. Model Weaving Support for Migrating Software Artifacts from AUTOSAR 2.0 to AUTOSAR 2.1.Int. Conf. on Embedded Real Time Software (ERTS 2008), Toulose (France), 2008.

[23] M. Wimmer, A. Schauerhuber, W. Schwinger, H. Kargl. On the Integration of Web Modeling Languages: Preliminary Results and Future Challenges. Workshop on Model-driven Web Engineering (MDWE 2007), held in conjunction with ICWE, Como, Italy, 2007.

[24] J.C. Preciado, M. Linaje, F. Sánchez. Designing Rich Internet Applications with engineering methodologies. Proc. of the 9th IEEE Int. Symposium on Web Site Evolution. IEEE Computer Society, 2007, pp. 23-30.

