MDWEnet: A Practical Approach to Achieving Interoperability of Model-Driven Web Engineering Methods

Antonio Vallecillo[1], Nora Koch[2], Cristina Cachero[4], Sara Comai[3], Piero Fraternali[3], Irene Garrigós[4], Jaime Gómez[4], Gerti Kappel[5], Alexander Knapp[2], Maristella Matera[3], Santiago Meliá[4], Nathalie Moreno[1], Birgit Pröll[6], Thomas Reiter[6], Werner Retschitzegger[6], José E. Rivera[1], Andrea Schauerhuber[5], Wieland Schwinger[6], Manuel Wimmer[5], Gefei Zhang[2]

[1]Universidad de Málaga, Spain
[2]Ludwig-Maximilians-Universität München, Germany
[3]Politecnico di Milano, Italy
[4]Universidad de Alicante, Spain
[5]Technical University Vienna, Austria
[6]Johannes Kepler Universität Linz, Austria

mdwenet@pst.ifi.lmu.de

Abstract. Current model-driven Web Engineering approaches (such as OO-H, UWE or WebML) provide a set of methods and supporting tools for a systematic design and development of Web applications. Each method addresses different concerns using separate models (content, navigation, presentation, business logic, etc.), and provide model compilers that produce most of the logic and Web pages of the application from these models. However, these proposals also have some limitations, especially for exchanging models or representing further modeling concerns, such as architectural styles, technology independence, or distribution. A possible solution to these issues is provided by making model-driven Web Engineering proposals interoperate, being able to complement each other, and to exchange models between the different tools. **MDWEnet** is a recent initiative started by a small group of researchers working on model-driven Web Engineering (MDWE). Its goal is to improve current practices and tools for the model-driven development of Web applications for better interoperability. The proposal is based on the strengths of current model-driven Web Engineering methods, and the existing experience and knowledge in the field. This paper presents the background, motivation, scope, and objectives of MDWEnet. Furthermore, it reports on the MDWEnet results and achievements so far, and its future plan of actions.

1 Introduction

Model-Driven Engineering (MDE) advocates the use of models and model transformations as the key features in all phases of software development, from system specification and analysis over design to implementation and testing. Each model usually addresses one concern, independently of the rest of the issues involved in the construction of the system. Thus, the basic functionality of the system can be separated from its final implementation; the business logic can be separated from the underlying platform technology, etc. The transformations between models enable the automated implementation of a system right from the different models defined for it.

Web Engineering is a specific domain in which model-driven software development can be successfully applied [1]. Existing model-driven Web Engineering approaches (such as OO-H [2], UWE [3] or WebML [4]) already provide a set of suitable methods and tools for the design and development of most kinds of Web applications. They address different concerns using separate models (navigation, presentation, business logic, etc.) and come with model compilers that produce most of the application's Web pages and logic based on these models. However, most of these Web Engineering proposals do not fully exploit all the potential benefits of MDE, such as complete platform independence, or tool interoperability. In addition, these proposals also have some limitations, especially when it comes to exchanging models or expressing further concerns, such as architectural styles or distribution.

Recently, the OMG's Model-Driven Architecture (MDA) initiative [5] has introduced a new approach for organizing the design of an application into different models so portability, interoperability and reusability can be obtained through architectural separation of concerns. MDA covers a wide spectrum of topics and issues ranging from MOF-based metamodels to UML profiles, model transformations and modeling languages.

However, the effective integration with the already existing model-driven Web Engineering approaches has been only partially achieved. The most interesting issue is the interoperability of models and artifacts designed using the different existing development methods to enable the use of synergies. The vision is, at the end of a long way, to count on either one unified method based on the strengths of the different methods, or interoperability bridges (transformations) between the individual models and tools that would allow their seamless integration for building Web applications.

Many groups of the Web Engineering community share these objectives. Lively discussions took place at both Model-Driven Web Engineering (MDWE) workshops in Sydney (2005) and Menlo Park (2006). A small number of groups decided to reinforce discussions on workshops with a set of planned activities in order to get concrete solutions to the current problem of interoperability of model-driven Web Engineering approaches. The initiative is called **MDWEnet** and started its activities in December 2006. This paper provides an overview of the motivation and background of this initiative (Section 2), its scope and objectives (Section 3), activities (Section 4), and future plans (Section 5).

2 Background and Motivation

The growing interest in Model-Driven Web Engineering has produced quite a significant number of results, which have materialized into a concrete set of MDWE approaches. As mentioned above, they provide suitable methods and tools for the design and development of Web applications, but they also present some limitations. So far, each group is mainly working on progressively improving their own proposals in an independent manner, with the exception of a couple of bilateral collaborations. One alternative solution is provided by the possibility of making Web proposals interoperate, being able to complement each other, and to exchange models between the different tools. This is precisely one of the goals of MDWEnet.

The authors of this paper met for the first time in Munich in December 2006, with the objective of coordinating the current efforts being carried out by individual groups in the field of MDWE. They are members from five of the groups that work on these topics, including the UWE, OO-H and WebML teams from the Universities of Munich, Alicante and Politecnico di Milano, respectively. The other two groups are from the University of Malaga, and from a joint cooperation between the Technical University of Vienna and the Johannes Kepler University of Linz, contributing with their knowledge on frameworks, metamodels and model transformations in the Web field [6,7,8]. The intention is to harmonize their efforts in order to be more effective, to avoid duplicated work, and to align their targets and goals. The plan was to start with a small number of groups first, and then to open to the rest of the MDWE community as soon as the first results were tangible and could be shown.

Several discussions took place during the meeting, most of them being representative examples of the topics and issues of current interest to the MDWE community. First, the current activities and work in progress of each group were presented. Then, a SWOT (Strengths, Weaknesses, Opportunities, and Threats) analysis of the situation of MDWE in the fields of MDE and Web Engineering was conducted, to provide a clear picture of the context and the current position from where to start. The following sections describe these issues, because they not only can be of help to the MDWEnet group, but can also be interesting to the whole MDWE community.

2.1 Work in Progress

The following list shows the topics and issues that each individual group has recently addressed:

a) Addressing new concerns in Web application development
- Software architecture (OO-H)
- Personalization (OO-H), Adaptation (WebML)
- Workflows (UWE, WebML)
- Integration with external Web Services (WebML)
- Requirements (UWE)

b) Quality evaluation
- Effort estimation (OO-H, WebML)
- Usability logs for analysing usage patterns and validation of navigational designs (WebML)

c) Metamodel profiling and integration
- Definition of a global framework (Málaga)
- WebML profiles (various)
- Metamodel integration (Wien/Linz)

d) Other

- Semantic annotations (OO-H)
- Automatic client-side code generation (WebML)
- Test derivation of applications (WebML)
- Analysis, validation and verification of models (WebML, UWE)
- Use of Aspect-Oriented Software Development techniques (e.g., for adaptation/access control) (UWE, Wien/Linz)

2.2 SWOT Analysis

A SWOT analysis was conducted to gain a better understanding on the Strengths, Weaknesses, Opportunities and Threads of current MDWE practices and approaches. The results are very illustrative, and show a field with plenty of possibilities and opportunities to grow and provide interesting benefits to the Web Engineering community.

a) Strengths

- Tool-supported methods
  …that work in practice!
  Significant improvements on productivity
- Tested and validated by real industrial usage
  Large companies
  Many projects (both privately and publicly funded)
- Wide knowledge and experience in Web Engineering
- Many groups working on interesting and useful extensions (see Sect. 2.1)

b) Weaknesses

- Of those approaches not using OMG standards
  Use of proprietary notations (many customers don't like them)
  Tools not aligned with MDA (yet)
- Of those using UML
  Tool support (for modeling and code generation)
- No interoperability of models and tools between individual proposals
  No reusability of efforts and developments
  No "core competencies" approach possible
- Current Web modeling languages…
  …are model-driven to a limited extent (e.g., the majority of approaches have not defined their metamodels, do not rely on model transformations, etc.)
  …partly provide concepts for modeling customization but no comprehensive support
- Customization of functionality cannot be captured separately but is scattered across all levels of a Web application model

c) Opportunities

- Web Engineering is a domain where MDE ideas can be successfully applied
- There is a current need for MDWE solutions in industry
  Real interest from customers
  Research funds (National and European)
- There is an interest in academia
  Journals, conferences
- MDE and MDA are fashionable now
  Claimed to be supported by everybody (OMG, IBM, Microsoft, Customers, etc.)
  Model transformation languages are becoming mature
- There is a group of people willing to co-operate to make it work
  MDWEnet is a concrete example
- Use the repositories of previous projects for conducting empirical studies on performance, quality, etc.

d) Threats

- MDE/MDA fails to deliver because of
  No tool support
  Customer dissatisfaction or frustration (probably due to too high expectations)
- We fail to deliver because of
  Result is worse than individual proposals, or
  Resulting method, techniques and/or notation are too complex, Learning is too difficult, or usability is not good enough
  No real applications (very complex) can be built
- Real goals not addressed; they are
  Too academic, or
  Too pragmatic

3 Scope and objectives of MDWEnet

The **scope** of the MDWEnet initiative is the model-driven development of Web applications, using different methods and tools, while ensuring the interoperability of their artifacts and models.

The overall **objective** is to improve current practices and tools for the model-driven development of Web applications, by making use of the strengths of current model-driven Web Engineering methods, and the existing experience and knowledge in the field.

The way in which we decided to reach this goal is by investigating the interoperability of model-driven Web Engineering methods, i.e., by trying to explore how Web proposals could interoperate, be able to complement each other, and exchange models between the different tools.

Two clear phases in the process were distinguished: (1) proof of concept and validation; and (2) application of the interoperability approach.

The first phase is focused on investigating how this interoperability can be achieved at a basic level (i.e., over the fundamental set of elements and functionality that any MDWE method should cover), and on its validation for three MDWE methods: OO-H [2], UWE [3] and WebML [4]. This phase is based on an incremental and iterative process, starting from a very small set of features and functionality that the different methods should deal with, which are progressively extended until the basic functionality offered by any MDWE approach is covered.

Once we achieve the required interoperability between the individual methods at that basic level, the second phase will use a set of representative Web applications to progressively extend these modeling elements and features, being able to deal with both static and dynamic aspects of Web application design.

4 Activities

During the workshop different possibilities to achieve the objectives were discussed, as already mentioned, focusing on two options: to use or not to use a common metamodel. In order to be able to define precise actions, the **MDWEnet** group had to make a set of decisions related to the technologies and tools to be used for implementing the actions. Some of these decisions were not easy to make, as described below. A plan of concrete actions was defined, relying on a strong commitment of the teams of all groups.

4.1 Possibilities

In general, there are many ways to achieve these goals, especially in the MDE field—which is neither fully mature nor well established yet. For instance, we had the following choices for tackling the problem of the interoperability between different MDWE approaches.

- Taking the best of each approach and try to define an integrated approach (in a similar way in which the UML was originally defined)?
- Developing a common metamodel?
- Preserving the nature of each web method and try to concentrate the efforts to process transformations between models?

We decided to initially explore two possibilities, and, once we have some concrete results, to look back and decide based on the pros and cons of each one. These possibilities, together with their advantages and disadvantages (a priori) are as follows.

Option 1: Definition of a metamodel for each individual approach and of the transformations between the different metamodels.

- Assumptions
  - There exists no common metamodel, or

  - No agreement is reached w.r.t. a common metamodel, or
  - The common metamodel is not expressive enough, or
  - Transformations are possible between all individual metamodels
- Benefits/advantages
  - Individuality is respected
  - Tools are readily available
  - Zoos (model repositories) can be “easily” built and maintained to share models
- Disadvantages
  - Integration and interoperation are much more difficult
  - Sharing tools is complicated
  - Too many transformations required [$n(n–1)$]

Option 2: Definition of a common metamodel

- Assumptions
  - There exists a common metamodel
  - An agreement is reached w.r.t. such a common metamodel
  - The common metamodel is expressive enough
  - Transformations are possible to/from all individual metamodels
- Benefits/advantages
  - Integration and interoperation are easier
  - Sharing tools is possible
  - Core competencies (presentation/information/tools/…)
  - Less transformations between metamodels [$2n$]
- Disadvantages
  - Individuality is somehow lost
  - Too many assumptions
  - Interoperability conflicts between different proposals

Of course, none of these options is free from problems. For example, should the common metamodel be (a) just the *basics* of MDWE; (b) the *intersection* of the metamodels of all MDWE proposals; or (c) the *union* of all metamodels?

Regarding the notation to express the metamodels, should we use MOF, eMOF, Ecore, KM3, or other metamodeling languages?

This leads to a more delicate question, regarding the MDE approach to use. Should we go try to be compatible with the OMG approach (which means using MOF, UML, QVT, etc), the Microsoft approach, or other (e.g., use AMMA and the ATLAS way)?

This has also to do with the choice of the modeling tools, since they do not interoperate at present. This is another important decision, since the only way to be able to seamlessly exchange models and artefacts is by sharing a common modeling tool (such as Enterprise Architect, MagicDraw, etc.). And the same is true for the model transformation language and tool to use: QVT (Together), graph-based (AGG, VIATRA, ATOM3), or other (e.g., ATL).

4.2 Decisions

As aforementioned, we decided to explore the two options above: (1) to define and use individual metamodels and transformations between them; and (2) to define a common metamodel and transformations to/from the metamodels of the different proposals. The common metamodel will be defined as the union of all metamodels. The metamodeling language will be Ecore, and the MDE approach will be based on the ATLAS group initiative, i.e., using the ATL as model transformation language. For drawing models we agreed to use MagicDraw as modeling tool.

4.3 Plan of actions

Based on these decisions, a concrete plan of actions was set up. It was organized into two phases, the first one running for 6 months. The actions to be developed during the first phase focus on the definition of a common metamodel, on the specification of the metamodels of the three initial proposals (UWE, OO-H and WebML) and on the transformations between these metamodels.

In addition, the actions should achieve the preparation of a survey of existing MDWE approaches and a "map" of communities that work on topics closely related to Model-Driven Web Engineering. A second phase would build on the result of the first one, and would consist of the definition of a Web Engineering modeling ontology, the evaluation of existing Web Engineering modeling tool environments and their capabilities for integration. Another goal is to cooperate in teaching and research, e.g., sharing teaching material and the acquisition of funding for joint projects.

4.4 Results so far

Although there is still a long way to go, we already count on a set of results, which could be of interest to the MDWE community.

The first one is a Wiki web, used by the group as a collaborative platform. The Wiki allows the exchange of information, documents, models, and tools, as well as the development of joint work on the material. It fulfills also the role of a repository of all kind of interesting information on model-driven Web Engineering topics.

The Wiki also contains the results that the actions have produced. In particular, it includes a collection of information on funding opportunities, the specification of the common metamodel, core metamodels of OO-H, UWE and WebML and a set of example model problems.

5 Future plans

The current activities are limited to the proof of concept of a first approach of interoperability of three methods OO-H, UWE and WebML. We also restricted the number of issues the different methods should manage to a small set of basic features of Web applications.

We plan to extend the current state with further modeling elements of the selected methods in order to cover all static and dynamic model-driven aspects of Web applications. Further methods and experimental material for volunteers to conduct experiments on external quality of Web applications developed with Web Engineering methods will be provided as well. At a long term plan the vision of a fully integrated environment where modeling and generation of Web applications using any method would be possible.

References

[1] Nathalie Moreno, José Raúl Romero, and Antonio Vallecillo. An overview of model-driven Web Engineering and the MDA. In Luis Olsina, Oscar Pastor, Gustavo Rossi and Daniel Schwabe, editors, Web Engineering and Web Applications Design Methods, volume 12 of Human-Computer Interaction Series, chapter 12. Springer, 2007.

[2] Jaime Gómez and Cristina Cachero. OO-H Method: Extending UML to Model Web Interfaces. Idea Group Publishing, pp. 144–173, 2003.

[3] Nora Koch, Alexander Knapp, Gefei Zhang, and Hubert Baumeister. UML-based Web Engineering: An Approach based on Standards. In Luis Olsina, Oscar Pastor, Gustavo Rossi and Daniel Schwabe, editors, Web Engineering and Web Applications Design Methods, volume 12 of Human-Computer Interaction Series, chapter 7. Springer, 2007.

[4] S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai, and M. Matera. Designing Data-Intensive Web Applications. Morgan Kaufmann, 2002.

[5] MDA Guide V1.0.1, omg/03-06-01, www.omg.org/mda

[6] Andrea Schauerhuber, Manuel Wimmer, Elisabeth Kapsammer, Wieland Schwinger, and Werner Retschitzegger. Bridging WebML to Model-Driven Engineering: From DTDs to MOF. Accepted for puplication in IET Software, 2007.

[7] Andrea Schauerhuber, Manuel Wimmer, and Wieland Schwinger, Elisabeth Kapsammer, and Werner Retschitzegger. Aspect-Oriented Modeling of Ubiquitous Web Applications: The aspectWebML Approach. 5th Workshop on Model-Based Development for ComputerBased Systems: Domain-Specific Approaches to Model-Based Development, in conjunction with ECBS, Tucson, AZ, USA, March 2007.

[8] Manuel Wimmer, Andrea Schauerhuber, Wieland Schwinger, and Horst Kargl. On the Integration of Web Modeling Languages: Preliminary Results and Future Challenges. Workshop on Model-driven Web Engineering (MDWE), held in conjunction with ICWE, Como, Italy, July, 2007