GUEST EDITORIAL

Guest Editorial to the Theme Section on Model-Driven Web Engineering

**Geert-Jan Houben · Nora Koch ·
Gustavo Rossi · Antonio Vallecillo**

1 Introduction

Model-Driven Engineering (MDE) is becoming a widely accepted paradigm for the design and development of complex distributed applications. MDE advocates the use of models and model transformations as key artefacts in all phases of the software process, from system specification and analysis, to design, development and testing. Each model usually addresses one concern, independently from 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, and the business logic can be separated from the underlying platform technology, etc. Transformations between models enable the automated implementation of a system from the different models defined for it.

The Web Engineering (WE) community soon discovered that the key aspects of MDE (abstraction through modeling; separation of concerns by multi-viewpoint specification, and software development by model transformation) perfectly matched the principles and practices promoted by WE, and allowed Web application designers to address some of their

G.-J. Houben
Technische Universiteit Delft, Delft, The Netherlands
e-mail: g.j.p.m.houben@tudelft.nl

N. Koch
Ludwig-Maximilians-Universität München, and Cirquent GmbH,
München, Germany
e-mail: kochn@pst.ifi.lmu.de

G. Rossi
Universidad Nacional de La Plata, La Plata, Argentina
e-mail: gustavo@lifia.info.unlp.edu.ar

A. Vallecillo (✉)
Universidad de Málaga, Málaga, Spain
e-mail: av@lcc.uma.es

most critical challenges, for example, the need to cope with the constantly growing complexity and with the new requirements on Web applications, in the face of a rapid evolution of the supporting technologies and platforms. Further architectural concerns, such as adaptation or distribution, need to be modeled and taken into account into new Web systems, beyond the traditional content-navigation-presentation aspects addressed by classical WE proposals. There is also an increasing trend towards the incorporation of emerging technologies like the Semantic Web, especially within the scope of the Web 2.0, its related technologies and richer applications. Furthermore, current Web applications need to interoperate with other external systems, which require their integration with third party Web-services, portals, portlets, and also with legacy systems. Finally, many of the existing WE proposals did not fully exploit all the potential benefits of MDE, such as complete platform independence, model transformation and analysis, and metamodeling.

Several initiatives started during the last decade to successfully apply MDE concepts and techniques for the engineering of Web applications. However, due to the complexity and heterogeneity of aspects that cover Web applications (hypertext structure, content, layout, services, security, etc.), there are still many open questions related to this area of research.

In 2005, the first workshop on Model-driven Web Engineering (MDWE) was held in Sydney, Australia, in conjunction with the International Conference on Web Engineering (ICWE 2005). The goal of the workshop was to provide a discussion forum where researchers and practitioners on WE and MDE could meet, disseminate and exchange ideas and problems, identify some of the key issues related to the model-driven development of Web applications, and explore together possible solutions. The workshop has been running since then, trying to bridge the domains

of MDE and WE every year. Five editions of the workshop (MDWE'05, MDWE'06, MDWE'07, MDWE'09 and MDWE'10) were held in conjunction with the ICWE conference. In 2008, MDWE was held as a workshop of the MODELS conference, mainly to locate the workshop in Europe and also to establish contacts with a model-based and domain-independent-community such as the one that gathers every year around MODELS.

After the fifth edition of the MDWE workshop, once the community had significantly evolved, the authors of this guest editorial proposed the Software and Systems Modeling (SoSyM) journal to organize a theme section on "Model-Driven Web Engineering". The goal was to provide a representative sample of advanced research being conducted in this area. The selected papers presented in this theme section are the result of this effort. They provide an overview of current open issues and identify potential lines for further research.

2 Selected papers for this theme section

A call for papers was announced in 2009, targeted not only to previous authors and contributors of any of the previous MDWE workshops, but also to the Web Engineering, Models and MDWE communities in general. The response was very positive, with 21 abstracts initially submitted. Twelve of them became full submissions, and underwent a thorough peer review process, with two rounds of reviews. All submitted papers were refereed by at least three well-known experts in the field, which included Luciano Baresi, Jordi Cabot, Sven Castelyn, Tony Clark, Sara Comai, Serge Demeyer, Dirk Deridder, Joan Fons, Howard Foster, Piero Fraternali, Thomas Hettel, Alexander Knapp, Ivan Kurtev, David Lowe, Vicente Pelechano, Alfonso Pierantonio, Davide di Ruscio, Fernando Sánchez, Wieland Schwinger, Massimo Tissi, Juan Manuel Vara, Eelco Visser and Manuel Wimmer.

At the end of the review process, three papers were selected for this theme section:

– Gorka Puente, Oscar Díaz, Javier Cánovas and Jesús García Molina, in their paper "Harvesting Models from Web 2.0 Databases" present *Schemol*, a domain-specific language for extracting models out of databases. The authors argue that in the same way Model-Driven Engineering has been extensively used to obtain models from existing software artefacts (e.g., for modernizing or reengineering existing systems), it can also be used to extract models from data. Specifically they target extensive data repositories like those existing in Web 2.0 applications, such as for instance blogs and wikis. The paper also

shows how the *Schemol* language can be effectively used to bridge the gap between existing database schemas and metamodels, therefore allowing the conversion between data repositories and their corresponding data models.

– Danny Groenewegen and Eelco Visser, in their paper "Integration of Data Validation and User Interface Concerns in a DSL for Web Applications" present a solution for the integration of declarative data validation rules with the structural constraints defined by traditional user interface models in the domain of Web applications. This integration permits unifying syntax, mechanisms for error handling, and semantics of validation checks, and it allows Web application frameworks to offer a consistent interface for user interaction and data validation. With their proposal the authors aim at tackling the current problem of many Web modeling tools, which do not make all types of data validation explicit in their models, hence hampering full code generation and model expressivity. They use WebDSL, a domain-specific language for the definition of Web applications, as a proof of concept for their proposal.

– Davide di Ruscio, Antonio Cicchetti, Alfoso Pierantonio and Ludovico Iovino introduce in their paper "Managing the evolution of data-intensive Web Applications by Model-Driven Techniques" an approach to support the migration and evolution of data-intensive Web applications. They use model-differencing techniques to detect the modifications that a model has undergone during its lifecycle. The authors are also able to automatically derive programs capable of migrating and adapting those aspects which are not directly derivable from the source models, such as persistent data or page layout. The approach is validated with different Web application modeling languages, and, in particular, with beContent and WebML.

These papers contribute in different aspects to the field of Model-Driven Web Engineering, providing a deeper understanding of some of the issues that practitioners face when designing and developing Web applications using models and model transformations. The papers cover several important topics, showing how domain-specific languages can be effectively used for capturing essential concerns of Web applications, and how models can enable performing advanced tasks such as extracting data from legacy applications or managing the evolution of Web systems.

the reviewers for the timely manner in which they assisted in choosing and making suggestions to improve the selected papers. We are aware of all the time and effort involved in producing a constructive report. Finally, we would like to show our appreciation to the SoSyM editorial office, and, in particular, to Martin Schindler and Bernhard Rumpe, for the excellent support in preparing this theme section.