Guest editorial to the special issue on MODELS 2011

Jon Whittle · Tony Clark

Published online: 14 March 2013
© Springer-Verlag Berlin Heidelberg 2013

For the past 14 years, the MODELS conference has been the premier venue for the exchange of innovative ideas and experiences of model-based approaches in the development of complex systems. The conference series covers all aspects of model-based development for software and systems engineering, including modeling languages, methods, tools, and their applications. MODELS is universally recognized as one of the top conferences in software engineering research with an acceptance rate averaging 20 % in recent years.

In 16–21 October 2011, the MODELS conference was held in Wellington, New Zealand. The conference received 167 papers of which 34 were accepted for presentation by the programme committee, resulting in an acceptance rate of roughly 20 %.

Research in software and system modeling is now a relatively mature field. Like any mature field, however, it can be a good idea to encourage fresh thinking. Whilst not wishing to reduce the importance of solid incremental research, the conference this year asked participants to think ahead to what modeling would be like a decade hence. For this reason, the Program Chairs selected Modeling in 2020 as the theme for MODELS 2011. The theme was chosen to encourage new perspectives about the future role of modeling in complex systems engineering.

As usual, the contributions to the MODELS conference were of a uniform high quality and the authors of the best papers were asked to submit extended versions to a MODELS 2011 Special Issue of SoSyM. As described below, the

selected articles reflect the issues raised by the theme of the conference: what will modeling be like in the future.

Industrial-scale systems are often very large and even though model-based techniques employ abstraction in order to deal with size and complexity, such approaches must scale appropriately and offer acceptable performance. Therefore, a challenge for the next decade is to develop techniques that support large models. The article *A Repository for Scalable Model Management* by Pagán, Cuadrado, and Molina addresses this issue by describing how huge models can be managed in a database that integrates with the Eclipse Modeling Framework.

The early days of modeling were dominated by a small number of tools that were implemented as closed worlds. Such a situation is unattractive for a variety of reasons and has led to the development of tool-frameworks and tool-ecosystems. Whilst this addresses the problem of vendor lock-in and large unwieldy tools, it raises a problem of model synchronization whereby models held in multiple tools are kept in-sync by change propagation. A challenge for the next decade is to develop techniques for synchronization. The article *Model Synchronization Based on Triple Graph Grammars* by Hermann, Ehrig, Orejas, Czarnecki, Diskin, Xiong, Gottmann and Engel describes how graph grammars can be used to formally specify a synchronization framework.

Modeling languages are often defined using meta-models allowing model-based techniques to be applied at the language and tool level. This has become increasingly important over recent years as interest in Domain-Specific Modeling Languages (DSMLs) has grown. Unfortunately, there is no universally accepted precise basis for meta-models, which makes it difficult to analyse and compare different approaches and to build a firm foundation for modeling languages. A challenge for the next decade is to develop a formal basis for meta-modeling. The article *Automatically*

J. Whittle
University of Lancaster, Lancaster, UK
e-mail: j.n.whittle@lancaster.ac.uk

T. Clark (✉)
Middlesex University, London, UK
e-mail: t.n.clark@mdx.ac.uk

Reasoning about Metamodeling by Jackson, Levendovszky and Balasubramanian uses algebraic data types and constraint logic programming to define a basis for meta-models and shows how this basis leads to proofs about the meta-models and automatic test-case generation.

Industrial systems engineering is increasingly moving away from green-field software development and must address large-scale software maintenance activities. If modeling is to be applied on an industrial scale, it must cover the complete life-cycle from specification and design to reverse engineering and maintenance. Therefore, an important research direction for the next decade is to increase the scope of model-based techniques. The article *A Fine-Grained Analysis of the Support Provided by UML Class Diagrams and ER Diagrams During Data Model Maintenance* by Bavota, Gravino, Oliveto, De Lucia, Tortora, Genero and Cruz-Lemus describes the results of experiments on the ease of model comprehension and model modification in terms of modeling notations.

A specific maintenance activity is *refactoring*. The use of existing models in system design is attractive, because it reduces the cost of model development and can support an important Knowledge Management issue that retains design knowledge within an organization. Given that models must increasingly address issues related to the maintenance of systems, model-based refactoring is an important technique. Constraints are often used to add semantic information to models and, therefore, any refactoring must retain the meaning of the constraints. Therefore, in order to support industrial-scale system maintenance, refactoring techniques must be developed that allow the meaning of the models to be retained or at least changed in well-defined ways. The article *From Well-Formedness to Meaning Preservation: Model Refactoring for Almost Free* by Steimann describes formal techniques that support such refactoring.

As noted above, model-based techniques must support very large systems. As the size of models increase, their comprehension becomes a problem. A technique that is used to address this issue with respect to programming languages is *program slicing* where sub-programs are isolated with respect to supplied criteria. However, the rise of DSMLs introduces a problem that model slicing techniques differ from application to application. Therefore, modeling techniques in 2020 will support slicers generated as part of the language definition. The article *Kompren: Modeling and Generating Model Slicers* by Blouin, Combemale, Baudry and Beaudoux describes a language for expressing model slicers for any DSML.

Author Biographies

Jon Whittle is a full Professor and Chair of Software Engineering at Lancaster University. He is also a Royal Society Wolfson Merit Award Scholar. He has been working with Model-Driven Development for over 10 years, including stints at NASA Ames Research Center (California), George Mason University (Virginia), IIT Kanpur (India), and more recently at Lancaster University. He has been intimately involved with the MDD research community during this time, serving as the Chair of the Steering Committee of the MODELS conference from 2006 to 2008 and as PC Chair in New Zealand in 2011. He also serves on the editorial board of the *Journal of Software and System Modeling*. Jon's current interests are in empirically investigating what factors lead to success or failure with MDD in industry.

Tony Clark is a Professor of Informatics and Head of the Computer Science Department in the School of Science and Technology at Middlesex University. Tony has experience of working in both Academia and Industry on a range of software projects and consultancies. While at Marconi Research Ltd (1985–1994), he worked in both Software Engineering and KBS. From 1994–2003, Tony was an academic firstly at Bradford University and then at King's College, London where he worked on languages for OO specification and design including contributing to a range of industry standards through participation in the Object Management Group leading to the development of a novel approach to meta-modelling and language design exploited via a spin-out company which Tony co-founded in 2003 and served as Technical Director 2003–2008. Tony has been an editor for several special journal issues including IEEE Software and SoSyM. His current work addresses the development and analysis of models for Enterprise Architecture. Further information: <http://www.eis.mdx.ac.uk/staffpages/tonyclark/>.