Guest editorial for the special section on MODELS 2016

Jörg Kienzle¹ · Alexander Pretschner²

Received: 20 April 2019 / Accepted: 25 April 2019 / Published online: 7 May 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

1 Introduction

The MODELS conference series continues to be the premier venue for model-based software and systems engineering covering all aspects of modeling, from theory to languages and methods to tools and applications. MODELS 2016 took place in St. Malo, France, from October 2 to October 7, 2016, as the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems. A total of 118 papers were submitted to the ‘Foundations’ track of MODELS 2016, of which 28 were accepted, hence at a rate of 24%. The ‘Practice and Innovation’ track received 27 submissions, of which 10 were accepted, hence at a rate of 37%.

It has become a much valued tradition that authors of the best papers at each MODELS conference are invited to submit revised and extended versions of their papers for publication in a special section in SoSyM. The selection of these papers is based on input from the Program Committee and on the reception of the papers at the conference.

This special section presents the five articles that resulted from this invitation. Each article was subject to the full SoSyM review cycle, and authors received anonymous feedback in two rounds of reviewing from three reviewers who are experts in the field. As a result, each article has been thoroughly revised and substantially extended compared to its conference version.

2 Selected papers

Just like the MODELS 2016 conference program, the selected papers span a large spectrum of topics. The articles chosen for this special section cover topics ranging from

foundational typing considerations over concrete syntax for graphical languages over improvements in the use of models in development practice to applications.

The first article, *Unifying nominal and structural typing*, by Thomas Kühne, argues that different methodological uses of models, namely construction and explanation, lead to different flavors of typing, namely nominal and structural typing. Kühne characterizes these approaches, analyzes their trade-offs, and proposes a scheme that supports fluidly transitioning between them.

The second article is *Metamodel specialization for graphical language support*, by Audris Kalnins and Janis Bazdins. It observes that the need for standardizing graphical notations for general-purpose languages does not necessarily carry over to domain-specific languages and goes on to argue for making meta models not just define abstract, but also concrete graphical syntax. This is done by specializing meta models for the abstract syntax, and carries the special benefit that these special meta models can directly be used to build or configure editors.

The next two articles are concerned with the practice of model-driven engineering. The article *Enforcing fine-grained access control for secure collaborative modelling using bidirectional transformations* by Csaba Debreceeni, Gabor Bergmann, Istvan Rath, and Daniel Varró is concerned with one important aspect of the practice of model-based engineering: How can confidentiality and integrity constraints be modeled and enforced in collaborative modeling environments? The authors present a formal collaboration schema which guarantees specific correctness constraints and show how to adopt it to online scenarios with on-the-fly change propagation and the integration into existing version control systems for offline scenarios.

The fourth article, *Advanced prefetching and caching of models with PrefetchML*, by Gwendal Daniel, Gerson Sunyé, and Jordi Cabot starts with the observation that prefetching rules are often defined at the data level, making them hard to understand and maintain. Moreover, low-level prefetching and caching components are difficult to align with scalable model persistence frameworks, and are less

✉ Jörg Kienzle
Joerg.Kienzle@mcgill.ca

Alexander Pretschner
alexander.pretschner@tum.de

¹ McGill University, Montreal, Canada

² Technische Universität München, Munich, Germany

present in NoSQL databases, often used to store large models. The authors therefore propose a framework that executes prefetching and caching strategies over models.

Finally, the fifth article, *Handling index-out-of-bounds in safety-critical embedded C code using model-based development*, by Gunter Blache, reports on solutions to the problem of generating safe embedded code, in the sense of avoiding out-of-bound array accesses. The basis is models developed in the ASCET tool that describe control algorithms and face special challenges such as different variants. Blache describes several solutions and illustrates them with case studies from industrial practice.

3 Conclusion

Model-driven engineering continues to be a thriving area of foundational research and practice, with tremendous

progress on many diverse fronts. The MODELS 2016 conference once again witnessed this large and healthy variety, which is also reflected in this special section.

Acknowledgements We would like to thank the Editors-In-Chief of SoSyM, Bernhard Rumpe and Jeff Gray, for their continued support of the long and successful collaboration between MODELS and SoSyM. We also very gratefully acknowledge the work of the referees who spent countless hours producing feedback that greatly improved the submissions. We thank Martin Schindler from the SoSyM editorial office who has provided valuable assistance with the review process. Last but not least, we thank the authors for submitting their work and thus providing the material that this special section is made of.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.