Software and Systems Modeling (2023) 22:1393–1396
https://doi.org/10.1007/s10270-023-01108-2

GUEST EDITORIAL

Guest editorial for the special section on MODELS 2021

Shiva Nejati[1] · **Dániel Varró**[2,3]

Published online: 18 May 2023

1 Introduction

The MODELS conference series is the premier venue for model-based software and systems engineering covering all aspects of modeling, from languages and methods to tools and applications. MODELS 2021 took place online (originally planned in Fukuoka, Japan), from October 10 to October 15, 2021, as the ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems. A total of 113 papers were submitted to the conference. The Foundations track of MODELS 2021 received 81 submissions of which 18 were accepted, while the Practice and Innovation track received 32 submissions of which 13 were accepted. Together, both tracks had an acceptance rate of 27%.

It has become a much-valued tradition that authors of the best papers at each MODELS conference edition 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 14 articles that resulted from this invitation and review process. Each submission was subject to the full SoSyM review process, and authors received anonymous feedback from reviewers who are experts in the field. As a result, each article has been thoroughly revised and substantially extended compared to its conference version.

✉ Shiva Nejati
snejati@uottawa.ca

Dániel Varró
daniel.varro@liu.se; daniel.varro@mcgill.ca

[1] University of Ottawa, Ottawa, ON, Canada

[2] Linköping University, Linköping, Sweden

[3] McGill University, Montreal, Canada

2 Selected papers

We classify the 14 papers included in this special section into two categories, corresponding to the two tracks of the conference: (1) foundations for model-driven engineering (eight) and (2) practice and innovations in model-driven engineering (six).

The eight papers included in the first category provide insights on mechanisms and tools for collaborative modeling in virtual reality, machine learning based recommender systems, automated synthesis of assured components, model-based safety argumentation, model matching, execution trace analysis, assurance for autonomous systems and regression testing. The six papers in the category of practice and innovations in model-driven engineering describe solutions crossing various domains and applications such as automotive, robotics, transportation and low-code development. Altogether, the papers are representative examples of the diversity, depth and maturity of the current research in model-driven engineering. A summary of each paper is described below.

2.1 Foundations for model-driven engineering

The paper *Design and evaluation of a collaborative UML modeling environment in virtual reality* by Enes Yigitbas, Simon Gorissen, Nils Weidmann and Gregor Engels explores the potentials of using Virtual Reality (VR) technology to assist collaborative modeling with the Unified Modeling Language (UML) by developing a vR-based modeling environment. The paper compares the use of such environment with traditional collaboration for software design using conventional devices (e.g., laptops) within a use study. The results highlight that while the effectiveness and efficiency of VR technology have certain disadvantages, the user experience for remote collaboration is significantly increased.

The paper *MORGAN: a modeling recommender system based on graph kernel* by Claudio Di Sipio, Juri Di Rocco, Davide Di Ruscio, Davide and Phuong Nguyen provides an automated modeling assistant for developers in Model-

Driven Engineering (MDE) to help complete models and metamodels by proposing a recommender system based on graph kernels. Graphs are first extracted from training data using natural language processing techniques and subsequently fed into a recommendation engine based on graph kernel similarity metrics to provide predictions and recommendations on partially specified metamodels. The approach is implemented in the MORGAN recommender system, and its effectiveness is studied on three real-world datasets.

The paper *Synthesizing verified components for cyber assured systems engineering* by Eric Mercer, Konrad Slind, Isaac Amundson, Darren Cofer, Junaid Babar and David Hardin aims to improve cyber-resilience in an avionics context by a model-based approach that inserts high-assurance components such as filters and runtime monitors to the system design to block malicious data and to detect behavioral anomalies. Designers can use code contracts to formally specify high-assurance components. The synthesis and verification of such components are carried out by a semantics-preserving verified compiler while their behavior is validated by test contracts. The approach is implemented in the open source BriefCASE toolkit, which is evaluated in a case study on unmanned air vehicles.

The paper *The ForeMoSt approach to building valid model-based safety arguments* by Torin Viger, Logan Murphy, Alessio Di Sandro, Claudio Menghi, Ramy Shahin and Marsha Chechik presents a framework for engineers to help develop valid arguments and strategies for safety assurance cases. The framework offers model editors for different model types to precisely capture safety arguments on a high-level of abstraction and instantiate safety properties templates using model queries. Potential reasoning errors in assurance cases are mitigated by formalizing and validating them using the Lean theorem prover. The effectiveness of the framework is demonstrated on complex assurance cases taken from literature.

The paper *RaQuN: a generic and scalable n-way model matching algorithm* Alexander Schultheiß, Paul Bittner, Alexander Boll, Lars Grunske, Thomas Thüm and Timo Kehrer addresses the n-way model matching challenge to identify common elements when merging software variants or consolidating views in software product lines. The RaQuN framework provides a generic n-way model matching algorithm using heuristics and multi-dimensional search trees for finding match candidates along range queries. The experimental evaluation carried out on various model matching benchmarks shows that the RaQuN framework significantly improves the scalability of n-way matching compared to the state-of-the-art while delivering better-quality matching results.

The paper *Execution trace analysis for a precise understanding of latency violations* by Maysam Zoor, Ludovic Apvrille, Renaud Pacalet and Sophie Coudert aims to help understand the root cause of requirement violations in execution traces obtained from simulators in case of temporal properties involving latency. The authors present a precise latency analysis (PLAN) approach to classify transaction executions based on their impact on latency. For that purpose, a dependency graph is constructed from a hardware/software allocation model using model transformations. Then, the analysis highlights how specific hardware or software elements contributed to latency violations of an execution trace using the dependency graph. The relevance of the approach is demonstrated in the context of a case study of a European project.

The paper *MoDALAS: addressing assurance for learning-enabled autonomous systems in the face of uncertainty* by Michael Langford, Kenneth Chan, Jonathon Fleck, Philip McKinley and Betty Cheng investigates safety-critical systems with machine learning components where behavior is partly acquired in a training environment; thus, it can be uncertain for an untrained runtime context. The paper presents a model-driven approach to manage self-adaptive behavior of learning-enable systems to account for runtime contexts where the learned behavior cannot be trusted by monitoring and evaluating goal models at runtime. A proof-of-concept demonstrator of the MoDALAS framework is provided in an autonomous rover case study.

The paper *Efficient regression testing of distributed real-time reactive systems in the context of model-driven development* by Majid Babaei and Juergen Dingel proposes *MRegTest*, a model-level regression testing framework for distributed systems that facilitates deterministic replay of traces for detecting regressions. Given a system model captured in UML-RT notation with communicating state machines, the presented approach introduces various algorithms for (i) critical variable identification, (ii) execution selection and (iii) regression detection steps to ensure that existing functionality is not affected by changes. An experimental evaluation of the prototype implementation demonstrates the scalability of the framework when analyzing large UML models while requiring fewer number of recorded traces.

2.2 Practice and innovations in MDE

The paper *Involving users in the development of a modeling language for customer journeys* by Ragnhild Halvorsrud, Odnan Ref Sanchez, Costas Boletsis, and Marita Skjuve addresses the development of the Customer Journey Modeling Language (CJML), a domain-specific modeling language (DSML) for representing service processes from the end-user's perspective. While numerous methods for handling the technical aspects of developing DSMLs have been formalized, user needs and usability aspects are often addressed late in the development process and in an ad hoc manner. The paper highlights how the CJML, which targets a

diverse and heterogeneous group of users, was systematically improved using various user-centered design techniques in close collaboration with the target group. Their feedback was instrumental in refining and evolving the syntax and semantics of CJML.

The paper *Assessing the usefulness of a visual programming IDE for large-scale automation software* by Bianca Wiesmayr, Alois Zoitl and Rick Rabiser concerns with the challenges faced in developing industrial control software, which is typically designed by domain experts rather than software engineers. These experts often use visual programming languages based on standards such as IEC 61131-3 and IEC 61499, which apply model-based engineering concepts to abstract from hardware and low-level communication. However, there is limited empirical research on the usability and utility of model-based development environments for visual programming languages. The authors discuss common control software maintenance tasks and tool capabilities based on existing research, and demonstrate the implementation of these capabilities in the 4diac IDE. In addition, they improve the tool based on a user study, and evaluate the tool in a reassessment study involving industrial automation engineers. The paper concludes by providing general implications for developers of IDEs in the context of (visual) model-based engineering tools.

The paper *OSTRICH: A rich template language for low-code development* by Hugo Lourenço, Carla Ferreira, João Seco and Joana Parreira introduces OSTRICH, a strongly typed templating language for the OutSystems low-code platform, aimed at simplifying component reuse and customization. OSTRICH builds on metamodel annotations to ensure well-formed and consistent code across application layers. The authors present novel type safety verification for template definitions and template arguments, ensuring model consistency across application layers. They have implemented this template language in a prototype of the OutSystems platform and successfully ported nine of the top ten most used sample code fragments, thus improving the reuse of professionally designed components.

The paper *Model-driven design space exploration for multi-robot systems in simulation* by James Harbin, Simos Gerasimou, Nicholas Matragkas, Athanasios Zolotas, Radu Calinescu and Misael Alpizar Santana addresses the complexity of engineering multi-robot systems by introducing ATLAS, a novel model-driven approach that supports systematic design space exploration and robustness analysis of multi-robot systems in simulation. Existing approaches for developing multi-robot applications lack a systematic mechanism for capturing crucial aspects and assessing the robustness of these systems. ATLAS features a domain-specific language that enables modeling the architecture of the robotic team, its mission, and the specification of the team's intelligence. The authors evaluate ATLAS and demonstrate its effectiveness through three simulated case studies: a healthcare Turtlebot-based mission and two unmanned underwater vehicle missions developed using the Gazebo/ROS and MOOS-IvP robotic platforms, respectively.

The paper *Reasoning over time into models with DataTime* by Gauthier Lyan, Jean-Marc Jezequel, David Gross Amblard, Romain Lefeuvre and Benoit Combemale presents a framework called DataTime, which addresses the challenges of using models at runtime for adaptive systems and digital twins, particularly in seamlessly interacting with the past, present, and future states of a system. DataTime captures the state of the system according to both time and space dimensions, modeled as a directed graph with local states on both nodes and edges. The framework offers a unifying interface to query past, present, and future (predicted) states of the system. This interface provides i) an optimized structure of the time series capturing past system states, ii) the ability to obtain the latest available sensor values, and iii) continuous micro-learning over graph edges of a predictive model to query future states, either locally or more globally, through a composition law. DataTime has been developed and evaluated in the context of the Intelligent Public Transportation Systems in the city of Rennes, France.

The paper *Repository mining for changes in Simulink and Stateflow models* by Monika Jaskolka, Vera Pantelic, Alan Wassyng, Richard F. Paige and Mark Lawford investigates the maintenance of Simulink/Stateflow models, which are primary design artifacts in automotive controls Model-Based Development (MBD). The authors aim to develop these models in a way that facilitates likely changes without adversely impacting the design quality. To gain a deeper understanding of frequently performed changes, they analyze an extensive industrial software repository and its associated version control system. This analysis provides insights into actual changes made to Simulink/Stateflow models in automotive controls development over many years, and classifies these changes to suggest how specific model alterations may impact system evolution. This research offers valuable information for improving the maintenance and evolution of such models in the automotive industry.

Acknowledgements We would like to express our gratitude to the Editors-In-Chief of SoSyM, Bernhard Rumpe and Jeff Gray, for their unwavering support of the fruitful collaboration between MODELS and SoSyM. We are grateful to the general chairs of MODELS 2021, Zhenjiang Hu, Tomoji Kishi, and Naoyasu Ubayashi, and the MODELS steering committee chair, Silvia Abrahão, for their tremendous effort organizing the conference during those difficult times. We are thankful to the anonymous reviewers for their valuable contributions to the paper evaluation process and to the authors for submitting their work and diligently revising it based on feedback. Finally, we extend our appreciation to Martin Schindler from the SoSyM editorial office for his assistance throughout the review process.

Shiva Nejati is an associate professor at the School of Electrical Engineering and Computer Science at the University of Ottawa and Co-director of uOttawa's recently established IoT Lab (Sedna). Nejati has published more than 80 scientific papers and received eight best or ACM distinguished paper awards as well as a 10-Year Most Influential Paper Award from CASCON. She serves as an associate editor for IEEE Transactions on Software Engineering and was PC co-chair for SSBSE 2019 and ACM/IEEE MODELS 2021. She has more than 15 years of experience conducting research in collaboration with the IoT, telecom, automotive, aerospace, maritime and energy sectors.

Dániel Varró is a full professor at Linköping University, Sweden, and an adjunct professor at McGill University, Canada. He is a co-author of over 170 scientific papers with seven distinguished paper awards and three most influential paper awards. He serves on the editorial board of Software and Systems Modeling, and served as a program co-chair of MODELS 2021, SLE 2016, ICMT 2014, FASE 2013 conferences. He delivered keynote talks at numerous conferences (incl. CSMR, SOFSEM and SAM) and international summer schools. He is a co-founder of the VIATRA open-source model transformation framework, and IncQuery Labs, a technology-intensive Hungarian company.