Software and Systems Modeling (2023) 22:795–796
https://doi.org/10.1007/s10270-023-01097-2

GUEST EDITORIAL

Guest editorial for the theme section on modeling language engineering

Benoît Combemale[1] · Romina Eramo[2] · Juan de Lara[3]

Received: 22 February 2023 / Accepted: 23 February 2023 / Published online: 10 March 2023

1 Introduction

Software-intensive systems are becoming more complex, driven by the need to integrate across multiple aspects. Consequently, the development of such systems requires the integration of different concerns and skills. These concerns are usually covered by different domain-specific modeling languages, with specific concepts, technologies, and abstraction levels. This multiplication of languages eases the development related to one specific concern but raises language and technology integration problems at the different stages of the software life cycle (e.g., it is necessary to explicitly describe the different kinds of relationships that exist between the different languages used in software development).

To support effective language integration, there is a pressing need to reify and classify these relationships, as well as the language interactions that the relationships enable. Similarly, the proliferation of domain-specific modeling languages increases the need for effective and efficient techniques for engineering languages and their support infrastructures (transformations, analysis tools, editors, execution infrastructure, debuggers, etc.). Hence, software developers are faced both with the challenging task of engineering each separate modeling language and associated technologies and with the task of integrating the different languages from different concern spaces.

These issues affect several domains and target areas. For instance, cyber-physical systems require the integration of different concerns and skills belonging to multiple disciplines: Increasingly, this industry is evolving toward multidisciplinary engineering from different domains (including software engineering, electrical engineering, safety engineering, and mechanical engineering). Another example are the emerging low-code platforms that enable the development of fully functional applications and raise the level of abstraction by combining visual development techniques (i.e., models) and code generation.

This theme section on *Modeling Language Engineering* (MLE) focuses on on topics addressing the mentioned challenges, including:

- Tools and methods for engineering modeling languages.
- Composability and interoperability of heterogeneous modeling languages.
- Language integration challenges.
- Model and metamodel composition.
- Language-based socio-technical coordination.
- Heterogeneous modeling and simulation.
- Multi-language and multidisciplinary environment.
- Modeling language engineering for low-code platforms.
- Scalability aspects in modeling language design.

2 Selected papers

The theme section received 11 submissions, out of which two were selected for publication after a careful review process. These papers were complemented by an expert voice paper by Professor Mark van der Brand.

The paper *Safe reuse in modeling language engineering using model subtyping with OCL constraints* by Artur Boronat focuses on low-code software development using Domain-Specific Languages (DSL) and Model-Driven Engineering (MDE). Low-code software development is heavily reliant on software reuse. This work presents a semantic reuse technique based on model subtyping against metamodels to handle the proper model-based design of DSLs. The versatility of the model subtyping technique is illustrated with common use cases culled from the research literature.

✉ Benoît Combemale
benoit.combemale@irisa.fr

[1] University of Rennes 1, Rennes, France

[2] University of L'Aquila, L'Aquila, Italy

[3] Universidad Autónoma de Madrid, Madrid, Spain

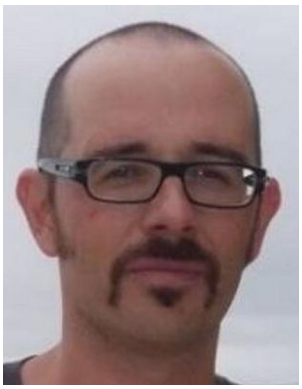
The paper *Advanced testing and debugging support for reactive executable* DSLs by Faezeh Khorram, Erwan Bousse, Jean-Marie Mottu, and Gerson Sunye focuses on Executable Domain-Specific Languages (xDSLs), and in particular on the definition and execution of behavioral models that accept external events and react by exposing events to the external environment. Since complex interactions may occur between the reactive model and the external environment, they should be tested early. This work proposes a generic testing approach for reactive xDSLs using the standard Test Description Language (TDL). The approach offers facilities to execute such test cases and to analyze them using two analysis tools: interactive debugging and mutation analysis.

The expert voice paper *A personal retrospective on language workbenches* by Prof. Mark van den Brand gives a great historical overview of the Dutch school about language engineering and more specifically the research path of the invited author. The paper offers a unique opportunity to understand the evolution of the research work in the software language engineering discipline, leading to concrete and seminal tools, as well as the emergence of the LDTA workshop series then the International ACM Conference On Software Language Engineering (SLE).

Acknowledgements We would like to thank the Editors-In-Chief of SoSyM, Bernhard Rumpe and Jeff Gray, as well as Martin Schindler from the SoSyM editorial office for all his support with the review process. We are also grateful to the anonymous reviewers for their invaluable contribution to the review of the papers, and to the authors for submitting their work and for working hard to revise it as advised.

Benoît Combemale is Full Professor of Software Engineering at the University of Rennes, France. He is evolving within the engineering school ESIR, and colead the research team DiverSE, joint to the CNRS Research Institute of Computer Science and Random Systems (IRISA) and Inria. He is also an adjunct researcher in the SM@RT team of the CNRS Research Institute in Computer Science of Toulouse (IRIT). He is interested in software engineering, including model-driven software and systems engineering (MDE), software language engineering (SLE), and software validation and verification (V and V); mostly in the context of (smart) cyber-physical systems and Scientific Computing. He is also the co-funding managing leader of the Eclipse GEMOC Initiative, supporting the development of the GEMOC Studio, an open-source cutting-edge language workbench. He has been the PC-chair of several conferences such as ECMFA, SLE, and ICT4S, and became recently a new Editor-in-Chief of the Software and Systems Modeling (SoSyM) journal.

Romina Eramo is Assistant Professor at the University of Teramo (Italy). Previously, she held a Ph.D. degree in Computer Science and different positions at the University of L'Aquila (Italy). Her research interests include model-driven engineering, software quality, continuous software engineering, DevOps, and digital twins. She is involved in several program committees of international conferences, reviewing activities, and conference organization. She published several articles in journals and proceedings of international events on her topics. She has been working and leading different European and Italian research projects. She is currently scientific coordinator for the AIDOART ECSEL-JU project.



Juan de Lara is Full Professor at the Computer Science Department of the Universidad Autónoma de Madrid, Spain. Together with Esther Guerra, he leads the modeling and software engineering research group (http://miso.es). His main research interests are in automated software engineering, model-driven development, domain-specific languages and language engineering, conversational interfaces, and augmented reality. This research has led to building many practical tools — including AToM^3, metaDepth, merlin, alter, DSL-comet, DSL-tao, and Capone — and the publication of more than 270 papers in international journals and conferences. He has been the PC co-chair of several conferences within his research areas, such as MODELS, SLE, ICGT, ICMT, and FASE, and has been involved in workshops on topics such as flexible modeling, multi-level modeling, and low-code development. He is on the editorial board of the Software and Systems Modeling (SoSyM) journal.