Softw Syst Model (2014) 13:171–172
DOI 10.1007/s10270-012-0254-z

GUEST EDITORIAL

Software and systems modeling with graph transformations theme issue of the *Journal on Software and Systems Modeling*

Andy Schürr · Arend Rensink

Published online: 20 June 2012

Over the years *model-based development* has rapidly gained popularity in various engineering disciplines. Numerous efforts have resulted in the invention of an abundance of appropriate modeling concepts, languages, and tools. Today modeling activities often span multiple disciplines and have to be addressed by collaborative efforts across disciplines such as industrial automation, business engineering, hardware/software co-design, real-time system development, Web 2.0 application design, and so forth. As a consequence, model-based development techniques related to the analysis, synchronization, and execution of families of models that are concurrently developed by different engineers on different levels of abstraction play a major role in many software and systems development projects.

Graphs, on the other hand, are among the simplest and most universal models for a variety of systems, not just in computer science, but throughout engineering and the life sciences. Graph transformations combine the idea of graphs as a universal modeling paradigm with a rule-based mathematically well-founded approach to specify their evolution. The area of *Graph Transformation*, also called Graph Grammars or Graph Rewriting, started around 1970 as a generalization of string grammars and term rewriting. It has since then grown to an independent research branch with its own Internal Conferences on Graph Transformation (ICGT), which is a forum for presenting both theoretical advances and practical applications. This conference series is complemented by the Symposium on Applications of Graph Transformation with Industrial relevance (AGTIVE) and a whole range of regularly organized satellite workshops of various computer science conference series.

Graph transformation concepts, languages, and tools today offer an appropriate framework for specifying and simulating as well as for predicting and verifying critical properties of models and model transformations for a whole range of different application domains such as bioinformatics, civil engineering, embedded and mechatronic system development, image generation and pattern recognition, and many others. The *Journal on Software and Systems Modeling*, therefore, invited the members of the graph transformation community to submit original, high-quality submissions for this theme issue focusing on foundations, applications, and tools that offer practical support for the enactment of model-based engineering processes while relying on graph transformation theory. The submitted papers underwent the usual multi-stage review process which in all cases involved experts both from the graph transformation as well as from the addressed application domain. Finally, seven papers were accepted for the publication in this issue. The first five of these papers focus on the application of graph transformation technology in a specific domain or for a specific purpose whereas the remaining two papers present new concepts for the manipulation of graph models.

- The paper *From Misuse Cases to Mal-Activity Diagrams: Bridging the Gap between Functional Security Analysis and Design* written by Mohamed El-Attar is concerned with modeling security requirements of software systems. Misuse-Case-based descriptions of system attacks are used as a starting point and translated into Mal-Activity diagrams. The presented translation process is an excellent example for a complex model or graph transformation.

A. Schürr (✉)
Technische Universität Darmstadt, Darmstadt, Germany
e-mail: andy.schuerr@es.tu-darmstadt.de

A. Rensink
University of Twente, Enschede, The Netherlands
e-mail: rensink@cs.utwente.nl

- The paper *A Transformation-Based Approach to Context-Aware Modeling* written by Sylvian Degrandsart, Serge Demeyer, Jan Van den Bergh, and Tom Mens highlights a different facet of the usage of model/graph transformation techniques in engineering. In this case transformation rules do not translate a higher-level static description of a software system into a lower-level description, but are used to specify the behavior of context-aware (self-adaptive) applications on mobile devices at runtime. Static analysis techniques are then employed to identify situations where the produced specification is ambiguous.
- The paper *Domain-Specific Discrete Event Modelling and Simulation using Graph Transformation* written by Juan de Lara, Ester Guerra, Artur Boronat, Reiko Heckel, and Paolo Torrini also deals with the specification of the behavior of software systems at runtime. In this case, concepts from the world of the discrete event system simulation community are combined with graph transformation concepts. An implementation that relies on the rewriting logic system Maude is presented, too. The resulting formalism can thus be used as a basis for the formal definition of other timed approaches or for direct prototyping purposes of event systems with dynamic communication relationships.
- The paper *A Fundamental Approach to Model Versioning Based on Graph Modifications: From Theory to Implementation* written by Gabriele Taentzer, Claudia Ermel, Philip Langer, and Manuel Wimmer belongs to a different category of submissions. It studies the general problem of merging possibly conflicting change sets of models for a selected modeling language. For this purpose, concurrently performed model modifications are translated into sets of sequences of graph modifications. Afterwards, operation- as well as state-based conflict detection and resolution techniques are studied. A prototype is implemented for the Eclipse Modeling Framework as technical space.
- The paper *Bridging the Gap Between Formal Semantics and Implementation of Triple Graph Grammars* written by Holger Giese, Stephan Hildebrandt, and Leen Lambers closes a gap between precise formalizations and efficient implementations of bidirectional model transformation approaches that rely on triple graph grammars. The authors present static analysis and runtime checks which guarantee that their batch implementation of triple graph grammars do not violate essential properties of their underlying formal definitions.
- The paper *GReTL: An Extensible, Operational, Graph-Based Transformation Language* written by Jürgen Ebert and Tassilo Horn introduces a kernel language for the implementation of graph transformations. GReTL both supports the definition of graph (transformation) algorithms in a simple domain-specific language as well as their implementation in Java on top of the GReTL library API. The kernel language and its realization thus can be used as a platform for the development of higher-level graph-based model transformation languages and tools.
- The paper *Graph and Model Transformation Tools for Model Migration* written by Louis M. Rose et al., finally, presents a model migration case study which is used to compare nine different model/graph transformation tools systematically. This case study has been proposed by the first author of the paper for the organization of the Transformation Tool Contest 2010. The other authors of the paper are representatives of the teams who participated in the contest. The presented comparison of the model migration specifications and the underlying transformation tools is used to highlight the state-of-the-art and motivate a research agenda related to future model transformation and, more specifically, model migration tool development.