

Guest editorial to the special section on model transformation

Zhenjiang Hu · Juan de Lara

Received: 23 October 2013 / Accepted: 4 November 2013 / Published online: 23 November 2013
© Springer-Verlag Berlin Heidelberg 2013

1 Introduction

Modeling is a key element in reducing the complexity of the development and maintenance of software systems. Software engineering paradigms like model-driven engineering (MDE) consider models as the primary elements in the software construction process. In this setting, model transformations are essential for elevating models from documentation elements to first-class artifacts of the development. Model transformation includes model-to-text transformation to generate code from models, text-to-model transformations to parse textual representations to model representations, model extraction to derive higher-level models from legacy code, and model-to-model transformations to normalize, weave, optimize, simulate, and refactor models, as well as to translate between modeling languages.

Model transformation encompasses a variety of technical spaces, including modelware, grammarware, and XMLware, a variety of transformation representations, including graphs, trees, and DAGs, and a variety of transformation paradigms, including rule-based graph transformation, term rewriting, and implementations in general-purpose programming languages.

The study of model transformation includes foundations, semantics, structuring mechanisms, and properties (such as

modularity and parameterization) of transformations, transformation languages, techniques, and tools. An important goal of the field is the development of high-level declarative model transformation languages, providing model representations of transformations that are amenable to 'higher-order' model transformation. To achieve impact on software engineering practice, tools, and methodologies to integrate model transformation into existing development environments and processes is required.

The International Conference on Model Transformation (ICMT) series started in 2008 as the premier forum for the presentation of contributions that advance the state-of-the-art in the field of model transformation aims to bring together researchers from all areas of model transformation.

The objective of this special section is to provide representative research papers in the field of model transformation, giving an overview of current open issues, and identify potential lines for further research.

2 Selected papers

This special section contains five papers whose shorter versions were presented at the 5th International Conference on Model Transformation (ICMT 2012), which was held during May 28–29, 2012 in Prague, Czech Republic. We selected 6 candidate papers out of 18 papers presented at ICMT 2012 for this special section based on the reviewers' comments and scores, and the quality of their presentations in the conference. The authors of these papers submitted an extended version that was considered for publication in this special section. The papers underwent a thorough peer-review process, and all papers were refereed by at least three well-known experts in the field. At the end of the review process,

Z. Hu (✉)
National Institute of Informatics, Tokyo, Japan
e-mail: hu@nii.ac.jp

J. de Lara
Universidad Autónoma de Madrid, Madrid, Spain
e-mail: Juan.deLara@uam.es

the following five papers were selected for this special section.

In “Reuse in Model-to-Model Transformation Languages: Are we there yet?” by Schoenboeck, Kusel, Wimmer, Kappel, Retschitzegger, and Schwinger, the authors provide a systematic and very in-depth survey of the existing transformation languages, in regard to their supporting language mechanisms for the reuse of model transformation. They give a nice taxonomy of the reuse mechanisms in model transformation, combining three dimensions, i.e., scope, specificity, and granularity. Under this taxonomy, they classified 15 mechanisms in total. The main part of this paper is a very detailed description of these mechanisms. For each mechanism, they give the definition, the illustrative examples, the transformation languages supporting this mechanism, and a further comparison of these languages.

In “Constraint-driven Modeling through Transformation,” by Demuth, López, and Egyed, the authors introduce a novel model transformation approach that relies on automatically derived model validation rules to specify the acceptable target models and informs the transformation engineer on the violating model segments. This way the engineer can manually define the required operations to be executed on the target model in order to fulfill all derived validation rules. The idea behind the approach is unique in the sense that it gives a complete tooling for automatically generating the instance level validation rules that are then fed into the ultrafast rule evaluation engine already developed by the authors. There are three impressive large case studies, explaining where transformation rules should be understood as constraints and showing the benefit of using constraints.

In “An Algorithm for Generating Model-Sensitive Search Plans for Pattern Matching on EMF Models,” by Varró, Deckwerth, Wieber, and Schürr, the authors address one of the crucial factors for the successful application of model transformations in practice: the efficiency of pattern matching. In particular, the work describes a model-sensitive plan generation algorithm to speed up graph pattern matching. The algorithm is based on dynamic programming, collects statistical data from the underlying EMF model, and uses the information for the optimization of the search plan.

In “Specification-Driven Model Transformation Testing,” by Guerra, and Soeken, the authors describe an advanced approach to automatic generation of input models and test oracles for model transformation testing. The generation is based on a formal specification of transformation properties (invariants, pre-, and post-conditions) in PaMoMo. After translating the properties in the specification into OCL constraints, input models are generated by composing an OCL expression from the properties the input model

should fulfill (according to certain coverage criteria) and then letting a constraint solver compute a valid input model that satisfies the specification OCL constraints, the meta-model integrity constraints, and the given OCL expression. A prototype tool supports editing of PaMoMo specifications and automates the generation of input models and test scripts from them. With three different case studies, it is shown that more exhaustive coverage criteria result in test suites with higher vigilance for less complete specifications.

In “Enhancing Classic Transformation Languages to support Multi-Level Modeling,” by Gerbig, Atkinson, and Tunjic, the authors adapt concepts of model transformation to multi-level models. Multi-level modeling is a generalization of the standard two-level meta-modeling setting where engineers are not forced to use one meta-level only, but models can spawn an arbitrary number of them. Multi-level modeling presents several benefits, but poses certain challenges when used together with model manipulation languages. In this paper, the authors address the challenges related to model-to-model transformations in a multi-level setting. In particular, they adapt the ATL transformation language for their work within the MelanEE multi-level modeling tool.

Acknowledgments We are grateful to the ICMT community for their excellent contributions to the ICMT conference and this special issue, as well as to the members of the ICMT Steering Committee for their helpful advice. We also thank the reviewers for their timely and accurate reviews. Finally, we would like to thank the SoSyM editorial office and in particular to Martin Schindler, for their continuous and excellent support.

Author Biographies



Zhenjiang Hu is a full professor of National Institute of Informatics (NII) in Japan. He received his BS and MS from Shanghai Jiao Tong University in 1988 and 1991, respectively, and PhD degree from University of Tokyo in 1996. He was a lecturer (1997–1999) and an associate professor (2000–2007) in University of Tokyo, before joining NII as a full professor in 2008. His main interest is in programming languages and software engineering in general, and functional programming, parallel programming, and bidirectional model-driven software development in particular. He is now serving on the steering committees of ICFP, Haskell, APLAS, ICMT, and BX, and is the academic committee chair of the NII Shonan Meetings. His e-mail address is hu@nii.ac.jp, and his Web page is <http://research.nii.ac.jp/~hu>.

parallel programming, and bidirectional model-driven software development in particular. He is now serving on the steering committees of ICFP, Haskell, APLAS, ICMT, and BX, and is the academic committee chair of the NII Shonan Meetings. His e-mail address is hu@nii.ac.jp, and his Web page is <http://research.nii.ac.jp/~hu>.



Juan de Lara is an associate professor at the Computer Science Department of the Universidad Autónoma in Madrid, where he leads the Modeling and Software Engineering research group (<http://www.miso.es>). He holds a PhD degree in Computer Science, and his research interests lie in meta-modeling, multi-level modeling, domain-specific languages, and model transformation. He has been a post-doctoral researcher at the MSDL lab (McGill University), the institute of theoretical computer science (TU Berlin), the department of computer science of the University of Rome “Sapienza,” and the University of York (UK). His e-mail address is Juan.deLara@uam.es, and his Web page is <http://www.ii.uam.es/~jlara>.

computer science (TU Berlin), the department of computer science of the University of Rome “Sapienza,” and the University of York (UK). His e-mail address is Juan.deLara@uam.es, and his Web page is <http://www.ii.uam.es/~jlara>.