Softw Syst Model (2006) 5:231–232
DOI 10.1007/s10270-006-0028-6

GUEST EDITORIAL

Guest Editorial to the Special Issue on Language Engineering for Model-Driven Software Development

Jean Bézivin · Reiko Heckel

Published online: 10 August 2006

Model-driven approaches to software development require precise definitions and tool support for modelling languages, their syntax and semantics, consistency, refinement and transformation. In order to support model-driven development for a variety of application contexts and platforms, efficient ways of designing languages are needed, accepting that languages are evolving and that tools need to be delivered in a timely fashion. In this respect, languages are not unlike software: A discipline of *language engineering* is required to support their design, implementation, verification and validation, delivering languages of high quality at low cost.

An important contribution of any engineering approach, besides the actual technology provided, is the meta knowledge about how different technologies are related and for which classes of problems they provide solutions. Well-known examples of such technologies, used by contributions in this special issue, include UML/MOF-based metamodelling, graph transformation, algebra and logic, data base technology, etc.

This special issue, published in two sections in this and the next volume of this journal, presents six contributions from two scientific events.

- The Dagstuhl Seminar 04101 on Language Engineering for Model-Driven Software Development.
- The Workshop on Graph Transformation and Visual Modelling Techniques (GT-VMT) 2004.

J. Bézivin (✉)
INRIA & University of Nantes, Nantes, France
e-mail: Jean.Bezivin@univ-nantes.fr

R. Heckel
University of Leicester, Leicester, UK
e-mail: reiko@mcs.le.ac.uk

The Dagstuhl Seminar discussed problems and solutions of language engineering in order to learn from examples presented by the participants and attempt a provisional classification and mapping. Three papers document the results of this work. Due to space limitations the first two, by Königs et al and Kühne, will be published in the next volume of this journal.

- In *MDI — a rule-based multi-document and tool integration approach* by Alexander Königs and Andy Schürr, the authors present a MOF-compliant model transformation language and tool based on triple graph grammars, in reply to the OMG's QVT (query, view, transformation) initiative.
- In *Matters of (meta-) modeling*, Thomas Kühne analyses basic notions of model-driven development such as "model" and "metamodel", contributing to the fundamental terminology of the domain by identifying different kinds of relations between models.
- *Definition and validation of model transformations* by Jochen Küster focusses on the analysis of model transformations for properties such as termination and confluence, making sure that transformations are well-defined.

The Workshop on Graph Transformation and Visual Modelling Techniques focussed on the foundations and applications of graph transformation, one of the underlying technologies of many model transformation approaches. In this special issue they are represented by three contributions.

- In *The design of a language for model transformations*, Aditya Agrawal, Gabor Karsai, Sandeep

Neema, Feng Shi and Attila Vizhanyo introduce a transformation language for model-to-model and model-to-code translations based on graph transformation and metamodelling.
- *Verifying object-based graph grammars: an assume-guarantee approach* by Fernando Luís Dotti, Leila Ribeiro, Osmar Marchi dos Santos and Fábio Pasini describes the use of graph transformation for modelling and verifying reactive systems using the SPIN model checker.
- In *Implementing a graph transformation engine in relational databases*,Gergely Varró, Katalin Friedl, and Dániel Varró introduce an efficient implementation for the transformation of large models using standard database technology.

In the design and implementation of modelling languages there is an increasing synergy between grammar-based and metamodel-based approaches. This is especially true for the emerging field of domain-specific transformation languages, to which many of the solutions in this special issue make a contribution.