

## Guest editorial to the special issue on MODELS 2008

Krzysztof Czarnecki

Published online: 4 May 2011  
© Springer-Verlag 2011

Welcome to this special issue of “Software and Systems Modeling,” devoted to selected papers of the eleventh ACM/IEEE International Conference on Model-Driven Engineering Languages and Systems (MODELS’08). MODELS is the premier annual conference featuring research results and practical experiences in model-driven engineering of complex, software-intensive systems.

MODELS’08 took place from September 28 to October 3, 2008, in Toulouse, France. This eleventh edition received a record number of 271 full paper submissions by authors from 40 countries. The accepted 58 papers spanned a wide range of topics, including model transformation, constraints, model management, domain-specific modeling, and modeling language semantics. The conference program also included three keynote presentations—“The Objects and Arrows of Computational Design” by Don Batory (University of Texas at Austin, USA), “Abstraction and Modeling—a Complementary Partnership” by Jeff Kramer (Imperial College London, UK), and “3D Parametric Models for Aeroplanes—from Idea to Design” by Patrick Rauhut (Airbus, Germany)—two panel discussions, and several workshops and tutorials.

This issue contains extended versions of five papers from MODELS’08. After the conference, the program committee nominated ten papers from the technical program to be considered for this special issue. The authors of these papers were then invited to submit extended versions of their papers. At least three experts reviewed each submission and each accepted paper underwent at least two rounds of revisions. I would like to thank all reviewers and authors for their hard work.

---

K. Czarnecki (✉)  
Department of Electrical and Computer Engineering,  
University of Waterloo, 200 University Avenue West,  
Waterloo, ON, N2L 3G1, Canada  
e-mail: kczarneck@gsd.uwaterloo.ca

The five papers contained in this issue reflect the comprehensive scope of MODELS.

“A Benchmark for OCL Engine Accuracy, Determinateness, and Efficiency,” by Mirco Kuhlmann, Lars Hamann, Martin Gogolla, and Fabian Büttner, proposes a benchmark for evaluating implementations of the Object-Constraint Language (OCL). The benchmark covers (1) the accuracy of implementing various OCL features, including standard operations and distinctive concepts such as ordered sets, (2) determinateness properties for OCL features with non-deterministic character, and (3) the efficiency of the implementation. The authors evaluate the benchmark by examining several OCL tools.

“MOOGLE: A Metamodel-Based Model Search Engine,” by Daniel Lucrédio, Renata P. de M. Fortes, and Jon Whittle, presents Moogole, a search engine for models. Moogole creates a rich search index and allows for complex queries by using metamodeling information. The evaluation shows that this information indeed improves the accuracy of the search results.

“Search-Based Model Transformation by Example,” by Marouane Kessentini, Houari Sahraoui, Mounir Boukadoum, and Omar Ben Omar, proposes an optimization-based approach for deriving model transformations from a collection of examples representing source-and-target model pairs. The key idea of the approach is to find transformation fragments in the examples that cover the best source model to be transformed. The authors compare two search-based algorithms, Particle Swarm Optimization and Simulated Annealing, for this purpose and evaluate the approach on the class-diagram-to-relational-schema transformation problem.

“Formal Foundation of Consistent EMF Model Transformations by Algebraic Graph Transformation,” by Enrico Biermann, Claudia Ermel, and Gabriele Taentzer, adapts the rich theory of graph transformation to the Eclipse

Modeling Framework (EMF), a popular technology for representing and manipulating models. In particular, the paper frames EMF model transformations as a restricted class of typed graph transformations, handling inheritance and containment. The authors demonstrate their approach using refactorings of statecharts as examples and describe the implementation of their concepts in a tool environment.

“Code Generation for a Family of Executable Modeling Notations,” by Adam Prout, Joanne M. Atlee, Nancy A. Day, and Pourya Shaker, describes semantically configurable code

generation for a family of behavioral modeling notations, which includes variants of statecharts, process algebras, petri nets, and SDL 88. The approach features a code-generator generator, supporting a wide range of semantic parameters and parameter values to cover the target family of notations. The authors compare the efficiency of the code generated by their approach with the efficiency of the code generated by commercial-grade code generators.

I hope that you will enjoy reading these extended papers from MODELS'08.