

Michał Śmiałek (Ed.)

# Promoting Software Modeling through Active Education

Educators Symposium  
at the 11<sup>th</sup> International Conference, MODELS 2008  
Toulouse, France, September 29, 2008  
Proceedings



Warsaw University of Technology  
Chair of the Theory of Electrical Engineering  
and Applied Informatics

All rights reserved; no part of this publication may be translated, reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. The quotation of those designations in whatever way does not imply the conclusion that the use of those designations is legal, without the consent of the owner of the trademark.

© by the authors, 2008  
Printed in Poland

ISBN 83-916444-8-0

## Preface

This volume contains the final versions of papers accepted for presentation at the Educators Symposium being part of the MODELS conference, held in Toulouse, France on September 29, 2008. The reviewing process was thorough and the authors of each submitted paper received very detailed comments on their work. At least three Program Committee members reviewed each paper. During the review process, a total of 9 papers were accepted for inclusion in the proceedings.

The symposium was organized around three types of submissions: full papers, short position papers and informal teaching technique descriptions. Formal submissions resulted in classical paper presentations and associated discussion. Informal submissions resulted in “tricks and tips” sessions where several educational artifacts were presented and discussed. The selected paper submissions formed three thematic groups: teaching model semantics, tool support for teaching models and modeling course concepts.

Within the first area, there were presented approaches to show to the students that modeling (specifically in UML) is more than just drawing nice (meaning also: syntactically correct) diagrams. Within the proposed courses, the role of modeling language semantics is presented to the students. This is important, as only presenting model semantics in a pedagogically effective way can supply the students with arguments for using models in practice. The second group of papers stresses the role of tools that support teaching. This does not only pertain to the obvious role of CASE tools. Novel approaches use tools to give certain pedagogical feedback on the models created by students. While creating their models within CASE tools, students receive instant comments that help in improving their modeling style. Such tools relieve the teachers and help in organizing courses with many attendants. They can also help in preparing fair and transparent criteria for assessing models created by many students. The final group of presentations treated new course ideas. This includes ideas on how to cope with large “volumes” of students (up to 1000 per course) and approaches which bring the modeling theory closer to practice by introducing project assignments that emulate real life in a software development organization.

The symposium concluded with discussion in groups formed around the above three thematic areas. The discussion concentrated on finding more effective ways to promote the MDE paradigm through software modeling education. The detailed results of this discussion can be found on the symposium web page: <http://www.iem.pw.edu.pl/edusymp08/>.



## Organization

The Educators Symposium has been organized as part of the MODELS conference and took place in Toulouse, France on September 29, 2008.

### General Chair

Michał Śmiałek (Warsaw University of Technology, Poland)

### Program Committee

Thomas Baar, akquinet tech@spree, Germany  
Jean-Marie Favre, IMAG, France  
Holger Giese, Hasso Plattner Institute, Potsdam, Germany  
Rogardt Heldal, Chalmers University of Technology, Sweden  
Øystein Haugen, SINTEF; University of Oslo, Norway  
Kai Koskimies, Technical University of Tampere, Finland  
Ludwik Kuzniarz, Blekinge Institute of Technology, Sweden  
Jerzy Nawrocki, Pozna University of Technology, Poland  
Lars Pareto, IT University of Gteborg, Sweden  
Pascal Roques, Valtech Training, France  
Jean Louis Sourrouille, INSA Lyon, France  
Miroslaw Staron, IT University of Gteborg, Sweden  
Perdita Stevens, University of Edinburgh, UK  
Michał Śmiaek, Warsaw University of Technology, Poland  
Georgi Tuparov, New Bulgarian University, Bulgaria  
Stephan Weibelzahl, National College of Ireland, Ireland  
Frank Weil, Motorola, USA  
Paula Filho Wilson de Padua, UFMG, Brazil



# Table of Contents

## Teaching Model Semantics

The UML is more Than Boxes and Lines . . . . .	1
<i>Yvan Labiche</i>	
Teaching Touchy Transformations . . . . .	13
<i>Martin Gogolla</i>	
Specifying Consistent Subsets of UML . . . . .	26
<i>Jean-Louis Sourrouille, Mohammed Hindawi, Lionel Morel, and Régis Aubry</i>	

## Tool Support for Teaching Models

Design of Pedagogical Feedbacks in a Learning Environment for Object-Oriented Modeling . . . . .	39
<i>Dominique Py, Mathilde Alonso, Ludovic Auxepaules, and Thierry Lemeunier</i>	
Automatic Checklist Generation for the Assessment of UML Models . . . . .	51
<i>Tom Gelhausen, Mathias Landhäußer, and Sven J. Körner</i>	

## Modeling Course Concepts and Innovations

Teaching Models @ BIG: How to Give 1000 Students an Understanding of the UML . . . . .	64
<i>Marion Brandsteidl, Martina Seidl, Manuel Wimmer, Christian Huemer, and Gerti Kappel</i>	
Project-based Teaching for Model-Driven Engineering . . . . .	69
<i>Mireille Blay-Fornarino</i>	
Collaboration between Industry and Research for the Introduction of a Model-Driven Engineering Course . . . . .	75
<i>Séverine Sentilles, Florian Noyrit, and Ivica Crnkovic</i>	
Improved Understanding through Complete Object-oriented Software Development Example: An Experience . . . . .	81
<i>Hazleen Aris</i>	
<b>Author Index</b> . . . . .	90



# The UML is more than Boxes and Lines

Yvan Labiche

Carleton University, Department of Systems and Computer Engineering,  
Software Quality Engineering Laboratory  
1125 Colonel By Drive, Ottawa, ON K1S5B6, Canada  
labiche@sce.carleton.ca

**Abstract.** The Unified Modeling Language (UML) is now the de-facto standard for the analysis and design of object-oriented software systems. There is a general consensus among researchers and practitioners that the UML could have a stronger semantic content. However, even the semantics of the UML, as described for example as well-formedness rules in the UML standard documentation, is not very well-known to many practitioners. As a result, practitioners often perceive the UML merely as a graphic tool. This paper discusses the apprenticeship of the UML semantics and presents a pedagogical method to help students overcome their limited view of the UML language as merely a set of annotated boxes and lines and to allow them to discover UML semantics.

**Keywords:** UML, well-formedness rules, consistency, teaching, laboratory

# Teaching Touchy Transformations

Martin Gogolla

University of Bremen (D), CS Department, Database Systems Group

**Abstract.** This paper reports on a teaching unit on model development and model transformation. One example model is first developed and considered as the source of various possible transformations. These transformations are explained implicitly afterwards by showing the different target models obtained by the transformations. The source model and the target models each emphasize a particular aspect, and an appropriate teaching method is chosen in order to communicate central ideas in a well-understandable way. The chosen teaching methods stress active student participation in the development of models and transformations.

# Specifying Consistent Subsets of UML

Jean-Louis Sourrouille, Mohammed Hindawi, Lionel Morel, Régis Aubry

INSA Lyon, LIESP, Bât. B. Pascal, 69621 Villeurbanne, France  
{Jean-Louis.Sourrouille, Mohammed.Hindawi, Lionel.Morel, Régis.Aubry}@insa-lyon.fr

**Abstract.** While increasing progressively its expressive power, UML has become more and more difficult to read and understand, especially for beginners. To teach the whole UML is not possible, therefore teachers only deal with a subset of UML. We present a framework for defining precisely a consistent subset of a language, allowing everybody to define his/her own subset. An extended example shows a way to simplify UML sequence diagrams. Our approach use standard UML tools; models in the simplified language are fully compatible with UML; model mapping between metamodels is done automatically from specifications based on marking with stereotypes; constraints are automatically translated from the simplified language to UML.

# Design of Pedagogical Feedbacks in a Learning Environment for Object-Oriented Modeling

Dominique Py, Mathilde Alonso, Ludovic Auxepaules, and Thierry Lemeunier

Laboratoire d'Informatique de l'Université du Maine (LIUM), Avenue Laënnec,  
Le Mans, France  
{Dominique.Py, Mathilde.Alonso, Ludovic.Auxepaules, Thierry.Lemeunier}@lium.univ-  
lemans.fr

**Abstract.** This paper describes the design of the pedagogical feedbacks in Diagram, a learning environment for UML object-oriented modeling. The project is rooted in several years of teaching practice and actual diagrams produced by novice students. Diagram is based upon an interaction framework that supports the learner's metacognitive activity. It includes a diagnosis module which compares the student diagram with a reference diagram, and produces the list of the structural differences between these diagrams. We show that these structural differences can be mapped onto differences that are relevant from a pedagogical viewpoint. The differences that are noticed in the student diagram give rise to feedbacks of three kinds: indications, questions and suggestions. We illustrate the feedback elaboration process by giving an example of a student diagram with the diagnosis results and the messages generated by the learning environment.

**Keywords:** modeling, UML, interaction, feedback, diagnosis, metacognition

# Automatic Checklist Generation for the Assessment of UML Models

Tom Gelhausen, Mathias Landhäußer, and Sven J. Körner

Institute for Program Structures and Data Organization  
University of Karlsruhe  
76131 Karlsruhe, Germany  
{gelhausen, lama, koerner}@ipd.uka.de

**Abstract.** Assessing numerous models from students in written exams or homework is an exhausting task. We present an approach for a fair and transparent assessment of the completeness of models according to a natural language domain description. The assessment is based on checklists automatically generated by the tool SUMO $\chi$  we present. SUMO $\chi$  directly works on an annotated version of the original exam text, so no ‘gold standard’ is needed.

# Teaching Models @ BIG: How to Give 1000 Students an Understanding of the UML

Marion Brandsteidl, Martina Seidl, Manuel Wimmer,  
Christian Huemer, and Gerti Kappel

Business Informatics Group (BIG)  
Institute of Software Technology and Interactive Systems,  
Vienna Technical University  
{brandsteidl|seidl|wimmer|huemer|kappel}@big.tuwien.ac.at

**Abstract.** In this paper, we report our experiences on teaching the Unified Modeling Language in the large. More precisely, about 1000 computer science and business informatics students attend our course *Object-Oriented Modeling* each year. Requiring a profound understanding of the UML, many advanced courses like Software Engineering or Model Engineering build on the knowledge imparted by our course. In order to achieve our ambitious teaching targets, we establish personal mentoring despite the mass enhanced with e-learning facilities.

# Project-based Teaching for Model-Driven Engineering

Mireille Blay-Fornarino

University of Nice Sophia – Antipolis  
CNRS, I3S Laboratory, RAINBOW team  
Sophia Antipolis, France  
`blay@polytech.unice.fr`

**Abstract.** For many years, I have been teaching meta-programming, software modeling and more recently model-driven engineering (MDE). Such innovative concepts and environments are not straightforward to understand for students. In this short paper, I list a few issues I met and I propose a new project-based approach to teach meta-modeling paradigms.

# Collaboration between Industry and Research for the Introduction of Model-Driven Software Engineering in a Master Program

S verine Sentilles, Florian Noyrit, Ivica Crnkovic,

M lardalen University, School of Innovation, Design and Engineering  
Box 883, SE-721 23 V ster s, Sweden  
{severine.sentilles, ivica.crnkovic}@mdh.se, florian.noyrit@gmail.com

**Abstract.** Teaching of model-driven software engineering in a software engineering master program curriculum suits well with the maturity of the students and their ability to understand the concept. Still the education might fail if the teaching is not based on convincing arguments and relations to the research or practice. We are about introducing a model-driven engineering course in our software engineering master curriculum and our goal is to relate the teaching with both research results and industrial practice. In this paper, we describe our approach – how to utilize the research results and how to relate the course to the requirements from industry we have cooperation with.

**Keywords:** Model-Driven Engineering, MDD education

# Improved Understanding through Complete Object-oriented Software Development Example: An Experience

Hazleen Aris

Universiti Tenaga Nasional, 43009 Kajang, Malaysia  
hazleen@uniten.edu.my

**Abstract.** This paper describes an effort made in overcoming the problem faced by students in understanding the object-oriented software development approach. It begins with an investigation to identify the source of the problem, which leads to the inclusion of a complete object-oriented software development example that is traceable from the analysis phase to the implementation phase, into the teaching contents. An evaluation was then performed after the completion of the object-oriented software development teaching using the updated contents. Result from the evaluation shows that complete and traceable object-oriented software development example contributes to the improved students' understanding of the approach.

**Key words:** OOSD module, OOSD teaching

## Author Index

Alonso, Mathilde, 39  
Aris, Hazleen, 81  
Aubry, Régis, 26  
Auxepaules, Ludovic, 39  
  
Blay-Fornarino, Mireille, 69  
Brandsteidl, Marion, 64  
  
Crnkovic, Ivica, 75  
  
Gelhausen, Tom, 51  
Gogolla, Martin, 13  
  
Hindawi, Mohammed, 26  
Huemer, Christian, 64  
  
Körner, Sven J., 51  
  
Kappel, Gerti, 64  
  
Labiche, Yvan, 1  
Landhäußer, Mathias, 51  
Lemeunier, Thierry, 39  
  
Morel, Lionel, 26  
  
Noyrit, Florian, 75  
  
Py, Dominique, 39  
  
Seidl, Martina, 64  
Sentilles, Séverine, 75  
Sourrouille, Jean-Louis, 26  
  
Wimmer, Manuel, 64