Software Modeling in Education: The 6th Educators' Symposium at MODELS 2010

Martina Seidl[1,⋆] and Peter J. Clarke[2]

[1] Business Informatics Group, Vienna University of Technology, Austria
Institute for Formal Models and Verification, JKU Linz, Austria
Martina.Seidl@jku.at
[2] School of Computing and Information Sciences, Florida Int'l University, USA
clarkep@cis.fiu.edu

Abstract. The Educators' Symposium (EduSymp) yields a major forum for software modeling education. Traditionally collocated with the ACM/IEEE International Conference on Model-Driven Engineering Languages and Systems (MODELS), EduSymp offers a unique opportunity for educators to present and discuss innovative pedagogical software modeling approaches. In this paper, a short retrospective on the 6th edition of EduSymp hosted in Oslo is presented. The program was a manifold of activities including interesting and thought-provoking oral presentations, an interactive breakout-session, and a panel discussion.

1 Overview

Research and academic instruction complement each other and eventually, the results of research become the foundation of the curriculum in academic disciplines. Whereas in some academic disciplines the knowledge taught to students is well established for a long time, in other disciplines, such as computer science, the knowledge mainly used is the result of recent or ongoing research. The training of computer scientists requires both established and cutting-edge knowledge, which are indispensable in preparing scientists for the challenges of the professional world.

In the field of software modeling, which continues to be an emerging research field with many highly innovative and practically relevant technological advancements, knowledge rapidly gets outdated. The challenge for teachers in the field of software modeling is to communicate well-established basic principles to the students, as well as to educate them on the scope and limitations of the novel and exciting technologies being developed. The Educators' Symposium (EduSymp), collocated with the ACM/IEEE International Conference on Model-Driven Engineering Languages and Systems (MODELS), offers a unique opportunity for educators to present and discuss innovative pedagogical software modeling approaches.

⋆ This work has been partly funded by the Austrian Federal Ministry of Transport, Innovation, and Technology and the Austrian Research Promotion Agency under grant FIT-IT-819584.

J. Dingel and A. Solberg (Eds.): MODELS 2010 Workshops, LNCS 6627, pp. 35–39, 2011.

Although many of the software modeling technologies are very mature from a conceptual point of view, the concrete implementations often provide several pitfalls to the users. Consequently, there are many questions educators are confronted with when developing software engineering curricula. A few of these questions are as follows:

- The field of software modeling continues to change rapidly. At what point in the continuum of change shall we start to teach modeling?
- Do students consider models to aid in software development or are they viewed as pretty pictures only?
- Can the benefits of modeling be realized without having proper tool support?
- Are tools imposing an extra inhibition/threshold in teaching modeling?
- Is it necessary for modeling tools to conform to standards or is it more important that they provide simplified concepts tailored for didactical purposes?
- Is it positive/negative when students are forced to use a specific tool implementation from a specific vendor? Do we teach them knowledge with an expiration date?

The 2010 EduSymp started with the thought provoking keynote "Formality in Education – Bitter Medicine or Bitter Administration?" given by Thomas Kühne[1]. The results of an online survey[2] conducted prior to the symposium offered an open forum for people interested and involved in software modeling education to discuss and exchange ideas on the questions previously listed.

The general consensus of the attendees at the 2010 EduSymp was that in software modeling education hands-on experience is extremely valuable for the students. The requirements for the modeling tools naturally vary based on the specific aims of the different courses (e.g., basic modeling, software engineering, model-driven engineering, and model engineering, among others). Overall it was thought that the available tools are mature enough, with respect to stability and documentation, that they may be successfully applied in software modeling courses. A huge point of criticism was their usability and user-friendliness which still poses a major burden to the students. In the context of these discussions many experiences and ideas were exchanged between the participants of the symposium.

This year's symposium was very well attended (between 20 and 50 participants attended the various sessions) which clearly indicates that software modeling education is an important issue within the modeling research community. We hope that the discussions initiated at the 2010 EduSymp will result in interesting and novel pedagogical ideas to support software modeling education. In addition, we expect the next edition of EduSymp will continue to increase the awareness on the importance of high quality education, and provide a forum for educators to meet, share, and discuss relevant issues in software modeling education.

[1] Associate Professor, School of Engineering and Computer Science, Victoria University of Wellington, New Zealand.

[2] A summary of the survey results is available at `http://edusymp.big.tuwien.ac.at/slides/survey.pdf`

2 Contributions

Seven papers (three long papers and four short papers) were selected to be presented at the 2010 edition of EduSymp covering a broad spectrum in software modeling education.

Teaching Model Driven Language Handling
Terje Gjøsæter and Andreas Prinz

Many universities teach computer language handling by mainly focussing on compiler theory, although MDD (model-driven development) and meta-modelling are increasingly important in the software industry, as well as in computer science. In this article, we share some experiences from teaching a course in computer language handling where the focus is on MDD principles.

The Role of User Guidance in the Industrial Adoption of AUKOTON MDE Approach
Jari Rauhamäki, Outi Laitinen, Seppo Sierla, and Seppo Kuikka

Model-Driven Engineering (MDE) has emerged as an actively researched and established approach for next generation control application development. Technology transfer to the industry is a topical research problem. Since most professional factory process control engineers do not have computer science backgrounds, there is an urgent need for studies of the role of user guidance in the professional learning, and thus, of industrial adoption of MDE approaches. In this study professionals were invited to a hands-on assessment of the AUKOTON MDE approach for factory process control engineering. Qualitative empirical material was collected and analyzed to identify the role of user guidance in the context of other factors impacting industrial adoption. Challenges in adoption that could be solved by user guidance were identified with the theory of organizational knowledge creation (SECI) model.

Implementation of the Concept-Driven Approach in an Object-Oriented Analysis and Design Course
Ven Yu Sien

As one of the most important tasks in object-oriented analysis and design (OOAD) is the abstraction of the problem domain into specific concepts or objects, information technology (IT) students need appropriate skills of abstraction in order to identify the essential concepts and relationships within a problem domain. However students in higher education generally find difficulty performing abstractions of real-world problems within the context of OOAD.

Concept mapping is a popular tool used in education for facilitating learning, comprehension and the development of knowledge structures. We have successfully adopted concept maps as stepping-stones to assist students in constructing class and sequence diagrams. In this paper, we present a framework for teaching

object-oriented (OO) modelling using concept maps. This framework comprising four teaching modules could be integrated into existing OOAD courses at the undergraduate or postgraduate level, and OOAD workshops to help software engineering educators resolve some of the difficulties they face in trying to teach OOAD. We also report results of an evaluative study on the effectiveness of integrating concept mapping techniques into an introductory OOAD course.

Teaching OCL Standard Library: First Part of an OCL 2.x Course
Joanna Chimiak-Opoka, Birgit Demuth

Our aim is to provide a complete set of materials to teach OCL. They can be used in bachelor or master programs of computer science curricula and for training in an industrial context. In this paper we present the first part of the course related to the OCL Standard Library. This part provides model independent examples to teach OCL types and their operations. It enables users to gain a basic understanding of the OCL Standard Library, which can be used as a starting point to write model constraints (OCL specifications) or model queries. Additionally, to the content of the paper, we provide a set of OCL packages, exercise proposals and lecture slides.

Role Allocation and Scheme in Software Engineering Course-Project
Ghafour Alipour

Role is a set of behavioural actions and responsibilities one takes in a specific situation. We all have many roles in our life. Differences among the role definitions reach mainly from the different emphasis of the software development method itself. Agile software development methods define roles to enhance communication and to produce a better product. In this paper, we describe a role allocation and scheme method whose data were gathered in students projects in software engineering course in the university.

New Media in Teaching UML in the Large - an Experience Report
Marion Brandsteidl, Konrad Wieland, and Christian Huemer

Huge classes with more than 800 students pose a major challenge to lecturers as well as to students, especially when a practical part is included. In order to successfully master lectures of this size, novel kinds of teaching media provide a multitude of enhanced opportunities. In this paper, we present our experiences with the application of new media in our undergraduate course Introduction to Object-Oriented Modeling (OOM). In this course, we teach approximately 800-1000 students per year the principles and techniques of UML 2.0. New media, i.e., technologies other than the traditional blackboard presentation like a document camera, web-based self assessments, or lecture recordings, are applied to support both, students and lecturers when learning and teaching, respectively. We empirically underline the acceptance of our concept with the feedback of our students concerning the newly used technologies gained through an extensive survey.

m2n: Translating Models to Natural Language Descriptions
Petra Brosch and Andrea Randak

To describe the structure of a system, the UML Class Diagram yields the means-of-choice. Therefor, the Class Diagram provides concepts like class, attribute, operation, association, generalization, aggregation, enumeration, etc. When students are introduced to this diagram, they often have to solve exercises where texts in natural language are given and they have to model the described systems. When analyzing such exercises, it becomes evident that certain kinds of phrases describing a particular concept appear again and again contextualized to the described domain.

In this paper, we present an approach which allows the automatic generation of textual specifications from a given Class Diagram based on standard phrases in natural language. Besides supporting teachers in preparing exercises, such an approach is also valuable for various e-learning scenarios.

3 Program Committee

The papers presented at EduSymp have been selected based on the novelty of the ideas, the impact of modeling during software development education, and relevance to the topics of the symposium. All papers passed through a rigorous review process, each paper received at least three detailed reviews prepared by internationally renowned experts. The list of the International Program Committee is shown below:

- Jordi Cabot, École des Mines de Nantes, France
- Fábio Costa, Universidade Federal de Goiás, Brazil
- Gregor Engels, University of Paderborn, Germany
- Robert France, Colorado State University, USA
- Martin Gogolla, University of Bremen, Germany
- Jeff Gray, University of Alabama, USA
- Øystein Haugen, SINTEF, Norway
- Gerti Kappel, Vienna University of Technology, Austria
- Ludwik Kuźniarz, Blekinge Institute of Technology, Sweden
- Timothy Lethbridge, University of Ottawa, Canada
- Werner Retschitzegger, Johannes Kepler University Linz, Austria
- Jean-Paul Rigault, University of Nice, France
- Patricia Roberts, University of Westminster, UK
- Michal Smialek, Warsaw University of Technology, Poland
- Dániel Varró, Budapest University of Technology and Economics, Hungary