Educators Symposium at MODELS 2008

Michał Śmiałek

Warsaw University of Technology,
Warsaw, Poland
smialek@iem.pw.edu.pl

Abstract. Model-driven engineering is becoming increasingly popular in software development projects as it raises level of abstraction, thus improving our ability to handle complex systems. In many academic and industrial centers, software modeling has already been introduced into their curricula. Despite this, it seems that education does not yet support the modeling paradigm well enough thus limiting its acceptance as a mature method of developing software systems. The goal of this symposium was to find ways to change this situation. Specifically, the symposium sought ways of showing benefits of modeling in a way that is pedagogically effective and attractive to the students. It also tried to make recommendations for placing the modeling courses in the overall software development educational path, which should include not only UML fundamentals but also a demonstration of the importance and place of modeling in the overall path from business (environment) to software products.

1 Introduction

Modeling is an important part of any software development effort, even if the developers do not realize that they are in fact doing modeling by eg. scribbling quick diagrams on whiteboards. It allows developers to abstract over complex problems and create "visual maps" of the "territory" covered by the resulting programs. However, usually, models are used only informally to support programming with additional (often hand-written) "visual notes". Although it is fairly widespread, the approach of using models in a systematic way which includes automatic model transformation and code generation has not yet got through to industry to the degree which would satisfy the advocates of the approach. This might be caused by lack of high quality, pedagogically effective MDE courses. Assuming we are convinced ourselves, we need to convince the students – prospective software engineers – that MDE gives significant benefits over other approaches. The students need to experience these benefits and gain appreciation leading to usage of MDE in their day-to-day business. Convincing the students of MDE is not easy, as it is usually treated as a "nice to have" feature within a software engineering project. We should thus discuss ways to convince people within the project teams to treat the model-driven approach as a "must have" that introduces additional, necessary level of abstraction thus

K. Czarnecki et al. (Eds.): MoDELS 2008, LNCS 5301, pp. 906–907, 2008.

leveraging understanding of complex systems. We need to convince students who normally use eg. only Java IDEs that they also need modelling plug-ins which would not only give a visual map of their Java programs but also allow for better organization of the path from user requirements to code.

2 Organization, Selection and Thematic Areas

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. From the submitted formal papers only 8 could be selected, and a thorough review process was needed. 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/

Acknowledgement

I would like to thank all the members of the Program Committee for their help in selecting the most relevant papers which highly contributed to the symposium’s success. I would also like to thank them for valuable comments which resulted in improving this summary text.