UML@Classroom: An Introduction to Object-Oriented Modeling

Martina Seidl

Institute for Formal Models and Verification
Johannes Kepler University Linz, Austria

The textbook *UML@Classroom*, now available in English [8] and German [7], evolved from a first semester course on object-oriented modeling at the Vienna University of Technology. Object-oriented modeling is a mandatory course for first-year computer science and business informatics students. Up to 1,000 students are attending this course per year and they have a very heterogeneous background on modeling and programming ranging from complete beginners to experienced software developers. One of the main challenges is to provide a solid basis to the beginners and to offer further material interesting to the advanced students by covering both the theoretical background as well as practical modeling. Details on the course organization are given in [2–6]. Despite our comprehensive teaching material (lecture videos, annotated slides, various kinds of self-assessments), we were regularly asked for a book accompanying the lecture. While many books on UML and object-oriented modeling are available, there was no book on the market which fitted our specific requirements: mostly the books were either too advanced or too high-level for our course, too specialized, too complicated, without examples, or simply outdated.

UML@Classroom was written with the aim to give a quick, but precise introduction to the world of UML and to provide an overview of the most important UML concepts and their relationships. The focus of the book is set on object-oriented modeling whose basics are shortly recapitulated while introducing graphical notation and the syntax and semantics of UML. Illustrative and intuitive examples all stemming from the university domain show how the various concepts are individually used in the respective diagrams and how they are interrelated if a model consists of multiple diagrams.

We abstained from presenting UML in all details, but we focused on the most widely used subset. In particular, we introduced the use case diagram, the class diagram with the object diagram, the state machine diagram, the sequence diagram, and the activity digram as well as their interplay. We also indicated the steps to be taken to turn a diagram into code and show some practical applications. Without going into the details, we also mention that UML itself is based on a metamodel which in turn is based on a metametamodel and that there are many more concepts which are then covered in the more advanced literature. However, we do not show the UML metamodel because in our experience, the introduction of this additional layer easily confuses beginners. Furthermore, we do not discuss variation points of the UML standard. Finally, as we did not want to write a book on software engineering or model-driven engineering (cf.

for example [1]), we did not show how to model a complete running system with all steps from the early design phases to the running system.

In addition, we offer extensive teaching material on the webpage accompanying our book (see `http://www.uml.ac.at`). Teachers can download the slides we use in our course (in German and in English) which we published under the Creative Commons License. This license allows teachers non-commercial use and the adoption of our teaching materials. For students, we provide access to our self-assessments which can be accessed via our Moodle-server. In addition, we developed a webpage with a UML quiz offering many attractive features to the users like storing the current status or getting detailed reports on the results. With these services we hope to make learning UML more attractive in a playful way.

References

1. Marco Brambilla, Jordi Cabot, and Manuel Wimmer. *Model-Driven Software Engineering in Practice*. Synthesis Lectures on Software Engineering. Morgan & Claypool Publishers, 2012.
2. M. Brandsteidl, T. Mayerhofer, M. Seidl, and C. Huemer. Replacing traditional classroom lectures with lecture videos: an experience report. In *Educators' Symposium @ MODELS 2012*, pages 21–27. ACM, 2012.
3. M. Brandsteidl, M. Seidl, and G. Kappel. Teaching Models @ BIG: On Efficiently Assessing Modeling Concepts. In *Educators' Symposium @ MODELS 2009*, 2009.
4. M. Brandsteidl, M. Seidl, M. Wimmer, C. Huemer, and G. Kappel. Teaching Models @ BIG — How to Give 1000 Students an Understanding of the UML. In *Promoting Software Modeling Through Active Education, Educators' Symposium @ MODELS 2008*, pages 64–68. Warsaw University of Technology, 2008.
5. M. Brandsteidl, K. Wieland, and C. Huemer. Novel Communication Channels in Software Modeling Education. In *MoDELS Workshops*, volume 6627 of *Lecture Notes in Computer Science*, pages 40–54. Springer, 2010.
6. M. Scholz, P. Kaufmann, and M. Seidl. Making UML "hip": A First Experience Report on Using Modern Teaching Tools for Object-Oriented Modelling. In *EduSymp @ MODELS 2013*, volume 1134 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.
7. Martina Seidl, Marion Brandsteidl, Christian Huemer, and Gerti Kappel. *UML @ Classroom - Eine Einführung in die objektorientierte Modellierung*. dpunkt.verlag, 2012.
8. Martina Seidl, Marion Scholz, Christian Huemer, and Gerti Kappel. *UML @ Classroom - An Introduction to Object-Oriented Modeling*. Undergraduate Topics in Computer Science. Springer, 2015.