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.

1 Background

In 2006 the curriculum of the 5 different BSc. branches of computer science as well as the curriculum of business informatics at the Vienna Technical University changed. The course *Object-Oriented Modeling* (OOM) was introduced for all students of those studies and suddenly we were confronted with the organization of a lecture which is attended by about 1000 students per year. The preceding modeling courses had been much smaller and the exercises were supervised by student tutors who corrected and discussed the solutions with groups of two students. In the winter term of the first year of the new curriculum, we tried to adopt the old mode. We had 15 student tutors in charge of the exercise course. We experienced that this approach did not scale and was very hard to manage. It was very problematic to find enough students with the necessary expertise to hold the tutorials. The different experiences and skills of the tutors were always a subject of complaints. Furthermore, the 15 student tutors were far not enough—for the summer term we would have needed almost twice as much. Not only that it is almost impossible to find such a huge number of suitable and interested students, the overhead of the management of such a group is not negligible anymore.

So we decided to reorganize the course and established a mode as described in the following sections. After having taught UML for 3 semesters in this manner, we can sum up that we have found a way to handle this huge mass of students with reasonable effort without the loss of personal interaction.

2 Structure of the Course

Organization. OOM consists of three parts: the lecture, the lab, and the e-learning self-assessments. For all parts, the adminstration is handled via tuwel, a Moodle[1] adaption of the TU Vienna where we provide all resources like slides and exercise sheets, announce the results of the tests, and provide support via an online forum. We recommend the books *UML @ Work* [1] and *UML Glasklar* [2] for self-studying purposes, but most students solely use the lecture slides to prepare the exercises and to study for the tests.

The participation in OOM is awarded with 3 ECTS points resulting in a total workload of 75 hours for an average student. To obtain a positive grade, the following requirements are defined: the practical part has to be completed successfully and three tests have to be passed. The tests contain multiple-choice questions, as well as open modeling tasks and have a duration of 30 minutes each. One test might be repeated at the end of the term.

The Lecture is hold in a traditional manner. The UML basics are presented to an auditorium with the possibility of discussion. In OOM eight lecture units are offered, the attendance is recommended but not mandatory. Structural as well as behavioral modeling techniques are covered, introducing the five most relevant UML diagrams: class, sequence, state, activity, and use case diagram. The content of the lecture is the basis for the exercise courses.

The Practical Part. The students are divided into groups of 50 persons, and each of these groups meets 6 times for the discussion of the exercise sheets. For each exercise the assistent professor chooses one student who must present and explain his/her solution as well as answer questions about the theoretical background. To pass the course, the students have to solve at least 24 of the 36 exercises. The groups are not guided by student tutors, but by three academic staff members of the Business Informatics Group. Each of those three holds two complete units, whereas the units correspond to the topics of the lecture.

E-Learning. For all diagram types presented in the lecture, we additionally provide broad e-learning support via tuwel containing 82 multiple-choice questions and 36 open modeling exercises with solution and some modeling guidelines. An example is shown in Figure 1. Solving those e-learning exercises is not mandatory, but they help to get some modeling practice and they are also a starting point for fruitful discussion in our forum, which is actively used by numerous students. In summer term 2008 about 60 percent of the students used the multiple-choice questions for test preparation, but only about 30 percent worked through the other examples. Here we see high improvement potential for the e-learning part, and we currently develop more advanced e-learning solutions than the current ones.

[1] http://www.moodle.org

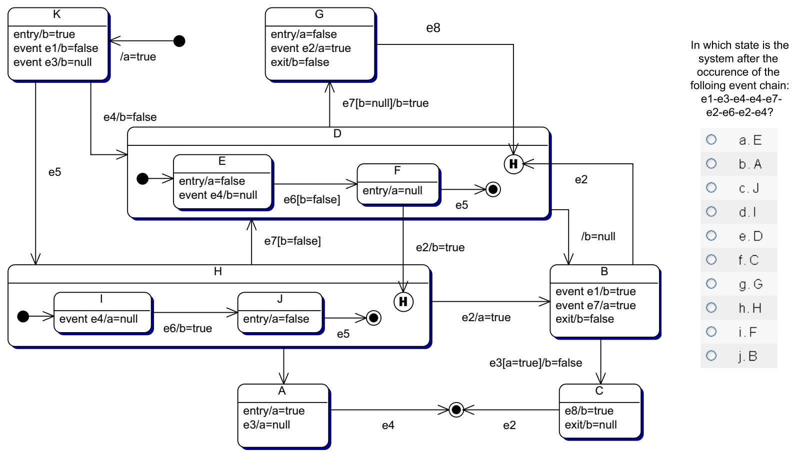

Fig. 1. E-Learning Example

3 Discussion

In this section we give an overview about the students' feedback about the course and we report the experiences we made teaching OOM for the mass.

Students' Feedback. The students have the opportunity to anonymously grade the courses they attend. A summary of the average grades for our modeling course since 2005 is shown in Figure 2 where the values may range from insufficient to excellent. The most remarkable curve concerns the evaluation of organization and support. With the annual increase of the number of students, the satisfaction on this issue decreased and reached a low in 2006 when the new curriculum was established. With the new mode, the situation deescalated and we are heading towards an "excellent"—although the number of students is still increasing. Also the contentment with the teaching material became much better since we started to provide the e-learning content. Furthermore, a slight growth concerning the practical applicability could be concerned, but at the same time the perception of the content's relevance slightly decreased. We plan to tackle this with more advanced e-learning facilities which will give a better link to practical applications. The most complaints concern the regular, mandatory exercise courses, but we observed an improvement of the open modeling examples at the tests. Interestingly, the students seem to prefer the 3 small tests to one big test.

Teachers' Experience. The new organization mode has the advantage, that it scales very well, and that the effort for the teachers is predictable. Overall, the mood in the exercise courses is very positive. Here we strongly try to avoid examination character, we encourage discussion, and we provide additional information to the models with special focus to modeling practices in real-world situations. In summer 2007, we tried a mode, were the presentation of the exercise solutions was voluntary, with the result that the students did not solve the

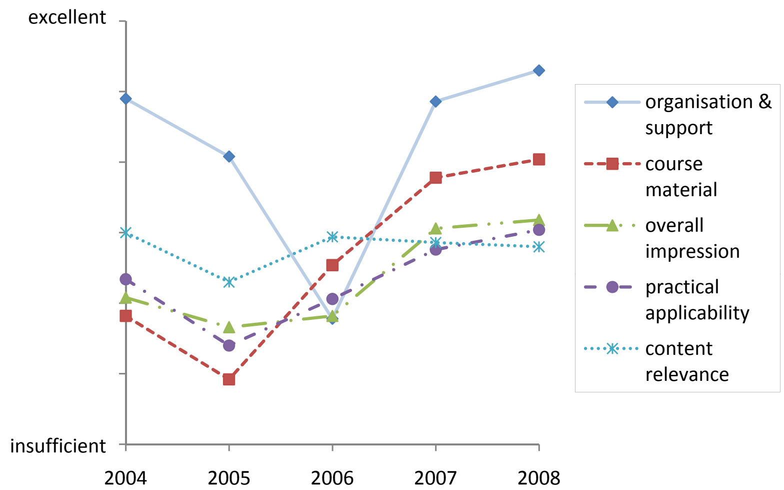

Fig. 2. Course Evaluation by Students

exercises at all, and no fruitful discussion was possible. Since then, it became mandatory to solve 75 percent of the exercises, and with the prepared students the exercise courses became highly interactive and also the results of the tests improved.

4 Advanced Modeling Course

BIG also offers an advanced modeling course—including a lecture and a lab—called *Model Engineering* which is obligatory for business informatics and optional for computer science master students.

Organization. Overall, we have about 150 students each term. The course covers principles, concepts, and tools, as well as practical experiences of Model Engineering. Therefore, lectures and exercises for the following topics are held: metamodeling, model transformations, code generation, and graphical modeling editors. Furthermore, the lecture also comprises advanced topics such as aspect-oriented modeling, MDE tools, and textual modeling languages. In the lab, the students have to cope with existing state-of-the-art MDE frameworks. The final output of the lab part is a modeling environment comprising a graphical modeling editor as well as model transformations and code generation facilities for producing running applications for a particular domain and technology out of models. As the OOM course, Model Engineering is also supported by the tuwel e-learning facilities, in particular for the lab exercises. For MDE frameworks, videos and tutorials are offered. They provide a starting point for the development of minimal metamodels, model transformations, code generation scripts, and graphical editors. The used frameworks are EMF for metamodel development, ATL for defining model transformations, oAW for developing code generators, and GMF for defining a graphical syntax for the metamodel.

Experience. In particular, for the running example in the lab, it is a must to give an example modeling language the students are familiar with. Thus, it is promising to build on knowledge about UML diagrams gained in the OOM course. In the past years, the students had to develop metamodels for subsets of UML diagrams, e.g., class, state, or sequence diagrams. When the students prepare solutions for the model transformations and code generation exercises, they are very glad to see how executable applications can be generated from UML diagrams. So to say, the Model Engineering course closes the gap between previous programming courses and modeling courses which are not connected appropriately in the bachelor studies, but are rather seen as competitive by the students. The only complains from students about the Model Engineering course concern the maturity of MDE technologies which are often not well documented and are sometimes not as user friendly as it is known from programming environments. However, with videos, tutorials, concrete suggestions, as well as forum support, we are trying to eliminate most issues about tooling problems in advance. Furthermore, it has to be noted that in the latest versions of the employed frameworks many issues have been eliminated.

5 Conclusion

Despite the huge amount of students in the UML course *Object-Oriented Modeling*, we offer broad mentoring on various levels—be it personal contact in the exercise courses, intensive support via forum and e-mail, examples in the e-learning system for self-study or discussions during the lectures and office hours.

OOM, scheduled for the first terms of the computer science and business informatics curricula is a necessary prerequisite for advanced modeling courses such as *Model Engineering*. To fully exploit the practical relevance of the UML, skills from other computer science disciplines, in particular programming skills, are essential. The students particularly recognize the value of modeling when they see the connection to executable applications and are therefore able to explore deeper semantic details of modeling languages compared to creating example models with pen and paper. Therefore, *Model Engineering* helps the students to gain an idea of the practical usage of modeling techniques in a broader context, whereas OOM teaches the basic concepts and the practical relevance only within the modeling level.

References

1. Martin Hitz, Gerti Kappel, Elisabeth Kapsammer, and Werner Retschitzegger. *UML@Work, Objektorientierte Modellierung mit UML 2.* dpunkt.verlag, Heidelberg, 2005.
2. Chris Rupp, Stefan Queins, and Barbara Zengler. *UML Glasklar. Praxiswissen für die UML-Modellierung.* Hanser Fachbuch, 2007.