

Learning aids and learners' activities in the field of object-oriented modelling

Torsten Brinda and Sigrid E. Schubert

Didactics of Informatics, University of Dortmund, Germany

Abstract: We introduce 'exploration modules' (EMs) as new aids for learning object-oriented modelling in learner-centred informatics education. The learners' activities, when using EMs are described. How far EMs can help learners in the solution of complex problems, for example, in modelling a library system is shown and discussed in detail. Moreover a phase concept for the inclusion of our overall conception for learning and teaching OOM for beginners, called "Didactic System for OOM" in teacher education is introduced, which has successfully been tested at the University of Dortmund.

Keywords: basic concepts, didactics of informatics, e-learning, learner-centred learning, learning environments

1. MOTIVATION

Secondary education has to prepare learners for the steadily changing everyday world and for life-long learning, in which complex informatics systems play an important role. The assessment of possibilities and risks of informatics systems has become a basic competence. Therefore learners have to understand the principles of analysis and construction of software, not only their usage. Object-oriented modelling (OOM) is a systematic method for software analysis and construction, which has been recommended in the overall plan for informatics education in Germany for this purpose by the German Informatics Society (Gesellschaft für Informatik, 2000). Now the 'Unified Modelling Language (UML)' and related tools for analysis and design are coming to be used in informatics education, but the

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35615-0_52](https://doi.org/10.1007/978-0-387-35615-0_52)

activities of learners are often still reduced to the application of standard software and problem-solving in a programming language, but product development and the education of software developers are not objectives of informatics education at secondary level. Modern educational concepts for secondary and university education stress the active role of the learners to prepare them adequately for self-paced, self-organised and life-long learning. Learning and teaching concepts, which in this context should be included more in education, are active, collaborative, peer and explorative learning (Turner, 1998). Suitable learning aids are necessary, which support learners in their explorative learning process. Scientific subjects, for example, use experimental installations for this purpose. In our department experiences were developed with software, with which learners can construct a part of the internet architecture in a closed environment and then check the data communication between network components by protocol simulation (Steinkamp, 1999). We found that such a software-based learning aid can enhance the understanding of working principles of informatics systems. It is now our aim to develop and to use such systems, so-called exploration modules (EM), for OOM (Brinda and Schubert, 2002).

2. LEARNER-CENTRED LEARNING

EMs support learner-centred learning by new learning-by-doing-oriented approaches to theoretical concepts, that should be learned in informatics education. With the help of EMs learners discover basic concepts of OOM and relationships between them. As a consequence, some teacher lectures become unnecessary. The learners take more responsibility for their own learning and for self-organisation. In this way they also become prepared for life-long learning. The explorative approach to theoretical concepts should not be mistaken for a trial-error-strategy. However, this requires systematic exploration strategies to be discussed before considering concrete exploration tasks and for the learners to have a rough pre-knowledge about OOM. Hypothesis formation of the learners is sharpened by the application of EMs, in giving them the possibility to first explore something, then manipulate it and finally check whether the change has caused the expected consequences. Reports of the EMs strengthen the error awareness of the learners. In traditional lessons it is hardly possible to discuss each learner's solution to a given problem. EMs can help, so more learners get individual reports. All these aspects increase the self-competence of the learners. The teacher designs the lesson as a discovery process. He demands the structuring of the individual findings of the learners and their linking up with theory. The application of EMs influences the quality of informatics

education in promoting communication and co-operation of the learners. In contrast to learning environments with an implemented learning strategy, learners must adequately identify problems and set up plans for obtaining findings. But learners are not left alone with their problems, because they become an integral part of the lesson conversation. Learning with EMs will not replace the traditional lesson. It will be an interesting alternative in suitable educational phases.

Exploration also promotes problem and situation-based learning, when correctly integrated in the informatics lesson. Van Weert classifies learning situations (van Weert, 2001). He distinguishes assignment, task, problem and situation-based learning situations. That learners can act reproductively and executively with the concept of exploration is taken for granted. In problem-based learning learners take up a *tactical role* by solving non-standard problems with adapted methods. The specification of desired results also belongs to this. In situation-based learning learners take up a *strategic role* by independently identifying both problems and solution methods. In the following the application of exploration in this context is discussed with concrete examples.

3. TACTICAL ROLE OF LEARNERS

In this example learners aimed to make the working processes in a library easier by using an informatics system. Pre-knowledge required was the static and dynamic basic concepts of OOM and the script language Python as the programming language. The EMs used here supported the library example. One of the first tasks was to divide the problem into sub problems.

At first the learners **explore model elements**. They discuss the problem and identify readers, books and the library as the essential objects. In the EM used here these library objects are clearly visualised as icons, with which simple interactions are possible, for example, simulation of lending events. The learners develop individual exploration plans to check their assumptions. In the team the findings of the learners are systematised. Typically one has to abstract from the application scenario to the object view. Object diagrams matching the current state of the application scenario can be generated and inspected. To understand the higher abstraction level 'class diagram', objects with common attributes that are computer-supported can be abstracted to a class. Here the list concept for data structuring and administration is needed, with books and readers being structured in lists. Then the learner can explore the concept of inheritance with the help of the similarly structured book and reader list. In the class view the EM therefore adds an abstract list class as a super class to book and reader lists. At first this class does not contain any attributes or methods. Therefore the learner

has the possibility to 'move' suitable attributes and methods between the classes along the lines of inheritance relationships. If the super class is an abstract class, then the semantics of the class model remain unchanged. At first it is recommended to 'move' attributes to the super class, to explore the combination of common class elements in a learning-by-doing-oriented manner. For the communication process the undoing of this abstraction step is sensible. Accompanying the explorative phases the group together constructs parts of the solution of the library problem. Assumptions of the learners regarding the problem-specific objects, attributes and methods are collected within the team and the findings are systematised together. In principle EMs are suitable for guided new discovery and unguided re-discovery of basic concepts. For better orientation and self-organisation of the learning process we recommend a 'didactic map' of knowledge structures, on which individual exploration paths between the basic concepts can be marked (Brinda and Schubert, 2001). The analogy with a map is helpful, because discovered and undiscovered knowledge areas can be structured clearly.

Secondly the **learners construct model views**. Here the EM places a precise textual description of the library problem at the learners' disposal. In it learners have as a first step the possibility to select class candidates in the text. After this selection they can decide, with check lists provided by the system, whether a class candidate is a class or not. Furthermore the EM provides the possibility to list class candidates according to selected heuristics. At first all the nouns in the problem description are offered as class candidates. Subsequently different heuristics can be activated (for example, if a noun can have a single value, it is an attribute candidate) to limit the set of candidates. Moreover the EM provides heuristics to identify further classes in the problem, which do not exist as nouns in the text. At every point it is possible to visualise the decisions made by representing the identified classes as symbols in a class diagram view. Attribute, method and relation candidates identified in the text are offered as building blocks for the class diagram. They can be combined and therefore promote the learning process of structuring. The learners construct individual class diagrams and discuss advantages and disadvantages of the variations in the group. A common diagram for further construction in the team is developed. Here the learners explore steps of a process model, that leads to a class diagram. To develop understanding of how in the finished library system objects cooperate to realise the system aim, it is necessary to visualise the usually hidden message exchange of co-operating objects. In this EM learners can, starting from a finished class diagram of the library, instantiate objects and construct an object diagram. By multiplicities, given object relationships can be generated by the system. An object diagram, constructed by a learner, can also be checked for its consistency with the given class diagram. In the

finished object diagram, learners can invoke methods of the objects and watch the message exchange with other objects and as their state changes.

4. STRATEGIC ROLE OF LEARNERS

For the consolidation of the learned material, learners solve a complex task, called a 'project' in informatics education. There are two kinds of projects: an *individual project* and a *team project*. After the successful conclusion of the library problem, each learner had the chance to choose a domain and to identify solvable problems. In the learning process no domain or content was given. The only restrictions were that aggregation relationships had to be used, the data structure 'file' had to be designed and the structure of the solution elements had to be enhanced with inheritance. The learners knew before the beginning of the development that the solution had to be presented by the group for the final evaluation. For example, the problems 'video management, that allows critiques to be saved of different persons to each video' and 'learning software for English education, that brings English active sentences into a passive form' were chosen by learners. Here they also profited from use of the EMs for the library problem. They also had to master the additional cognitive demand of the specialist transfer on the chosen problem. In this section strategic tasks are described, where exploration was helpful.

The **view change** plays an important role. The learners work on their chosen problems and for this purpose construct a class diagram (static model) and a sequence diagram (dynamic model) for each essential method. Driven by demand they use the EMs, which were discussed in Section 3. Since the learners had to identify the solutions of their individual problems more for themselves, the consistency and the completeness of their models plays a bigger role. To become aware of the consideration of consistency and completeness of models, they use an EM, with which they can construct a class diagram and sequence diagrams for their chosen problems. On demand, the system checks the consistency of both views, and shows missing elements. Method or class identifiers, which are used in a sequence diagram but not defined in the class diagram, are identified by the EM the same way the signatures of methods in the different views are compared and possible contradictions are disclosed. The learners evaluate the reports of the system and realise in which way the consistency and completeness of models can be ensured. With this system only the consistency and completeness on the basis of the learners' constructed models can be checked. Logical errors or forgotten classes cannot be identified by the system. For this purpose there are lesson phases, where the learners present the state of their work to the group, analyse and discuss it with them. The evaluation of the reflection of

the models of their classmates leads to findings about the solutions of their own problems. For learners the view change also has the purpose of combining static and dynamic sub-models to give an overall picture of the system and to overcome a known cognitive barrier. Therefore they use an EM, which visualises the linking of a class and sequence diagrams in the library scenario. If an object is produced in the sequence diagram by a constructor call, then the corresponding class constructor is emphasised in the class diagram. The same is so for called methods of a class. Every single method or class, used at least once in a sequence diagram, gets a marking. After the analysis of all sequence diagrams, the learner has the possibility to realise which classes and methods have not been used so far. A better picture of the interplay of static and dynamic models is achieved, which the learner transfers to the chosen problem.

EMs are usually used as learning aids, but they can also be studied as objects, as examples of complex **informatics systems**. For the solution of their chosen problems, the learners analyse EMs and their development documentation. Besides the software, this offers the problem description, analysis and design documents and the complete source code. Working with these documents requires well-founded knowledge about modelling techniques and programming and therefore can only take place in an advanced stage of informatics education. The group analyses the documents. Successful structured ideas are identified and generalised in the group. The learners reflect these findings and include them in the solution of their own problems. Learners also use the documentation to find out how certain model elements are implemented in the programming language. However, this is only possible if the EMs are implemented in the same programming language that the learners use for their own problem solution.

5. INCLUSION IN TEACHER EDUCATION

For inclusion in teacher education we developed a phase concept. It was successfully tried out in the education course of student teachers of informatics at the University of Dortmund.

In the **1st phase** student teachers discuss our theoretical conception called 'Didactic System for OOM' (Brinda and Schubert, 2001) in lectures and seminars. The focus is on the integration of the components 'exercise classes', 'exploration modules' and 'knowledge structures' in informatics education. In the field of the EMs, methods are discussed, how different exploration strategies of learners can be uncovered and analysed. Each EM can save all user interactions in a log file. Examples for user interactions are discussed, different interaction patterns are identified and assigned to cognitive processes (Tholander, 2001). This method, and typical errors that

are found, can help the student teachers in their own lessons to clarify the competence profiles of the learners. It is complicated by the fact that until now there have been no verified results about how knowledge about informatics in principle is built up in the learners' minds.

Teacher traineeships (2nd phase) are a compulsory but also very valuable study component for student teachers of informatics at the University of Dortmund. For one semester each student teacher has to prepare, hold and reflect on several informatics lessons together with mentors from the university and school. The other student teachers in the course document the lesson, when another student runs it, and later contribute to the reflection. In preparation of such teacher traineeships the student teachers use 'knowledge structures' for the structuring of the educational process. They develop 'exercise classes' for OOM, discuss suitable example contexts and derive concrete exercises for the lessons from both. In the lessons the learners' solution strategies for an exercise are explained afterwards, structured and combined with the abstract exercise class and used by the learners for similar exercises in the future. The student teachers analyse the EMs' educational success. For this they are introduced to the evaluation of educational processes by video analysis (Magenheim and Schubert, 2000) and also learn about difficulties. It is quite difficult to identify learning processes of the pupils on a video tape, where only one can watch or hear them or observe how they work with their computers. Moreover the pure existence of a video camera in the lesson can already change the behaviours of the learners and so falsify the picture. The results of the analysis of informatics lesson videos can be enhanced by the use of log files (see above). Then, however, one has to cope with a wealth of data and the question remains as to how far all of them are valuable.

To strengthen subject knowledge of software engineering education, the student teachers develop their own EM in a project (3rd phase). At the University of Dortmund the project is a study component, where two or more student teachers solve a large-scale problem in relation to informatics education by teamwork. They analyse available EMs and design a new one for a self-chosen informatics field. From this they get a deeper understanding of the assembly of EMs and they also sharpen their competences in object-oriented problem-solving. In extended essays student teachers work on their own, limited research task. For example they can prepare a short sequence of lessons using the 'exercise classes' together with mentors from the university and school (team-teaching). Afterwards the student teacher documents the prepared lessons when the school mentor holds them. These texts as well as solutions of learners to prepared exercises are analysed for typical errors, hints for the motivation of learners, etc. From this a deep understanding of educational processes can be achieved.

The University of Dortmund intends to explore the teacher education according to the Bachelor-Master programme concept. Then the conception introduced will have to be revised, because the Bachelor programme also has to achieve professional competences. Further conceptual and empirical research work will need to be undertaken in the future.

REFERENCES

- Brinda, T. and Schubert, S. (forthcoming) Didactic system for object-oriented modelling. The 7th World Conference on Computers in Education - Networking the Learner. Copenhagen, Post Conference Book. Boston: Kluwer Academic Publishers
- Brinda, T. and Schubert, S. (forthcoming) Exploration of Object-Oriented Models in Informatics Education. Conference on 'Social, Ethical and Cognitive Issues of Informatics and ICT' 22-26 July, 2002, Dortmund, Germany. Post Conference Book. Boston: Kluwer Academic Publishers
- Gesellschaft für Informatik e.V. (2000) Empfehlungen für ein Gesamtkonzept zur informatischen Bildung an allgemein bildenden Schulen. In LOG IN 20 (2000) 2, I-VII.
<http://ddi.cs.uni-dortmund.de/lehre/diplom/steinkamp>
<http://www.mmi.unimaas.nl/euro-cscl/Papers/162.pdf>
- Magenheim, J. and Schubert, S. (2000) Evaluation of Teacher Education in Informatics. In D. Benzie and D. Passey (eds.) Proceedings of the Conference on Educational Uses of Information and Communication Technologies – 16th World Computer Congress 2000, Beijing, China.
- Steinkamp, D. (1999) Informatik-Experimente im Schullabor. Diploma thesis. Department of Informatics, University of Dortmund.
- Tholander, J. (2001) Students Interacting through a Cognitive Apprenticeship Learning Environment. Euro-CSCL 2001, Maastricht, 22 - 24 March 2001.
- Turner, A. J. (1998) Trends in teaching informatics. In F. Mulder and T. van Weert (eds.) Informatics in Higher Education. London: Chapman and Hall
- van Weert, T. (2001) Co-operative ICT-supported learning. A practical approach to design. In R. Keil-Slawik and J. Magenheim (eds.) Informatikunterricht und Medienbildung. Bonn: Köllen

BIOGRAPHIES

Torsten Brinda is assistant professor of the group 'Didactics of Informatics'. His research interests are concepts for e-learning, and learning and teaching object-oriented modelling in secondary and higher education. **Sigrid Schubert** has taught informatics in secondary, vocational and higher education. She is professor in 'Didactics of Informatics' at the University of Dortmund. Her research interests are informatics in secondary and higher education, collaborative learning, e-learning, knowledge management, multimedia and teacher education. She is the German national representative IFIP Technical Committee 3 on Education.