Didactic System for Object-oriented Modelling

Torsten Brinda and Sigrid E. Schubert

Didactics of Informatics, University of Dortmund, D-44221 Dortmund, Germany
brinda@ls12.cs.uni-dortmund.de

Keywords: didactics, knowledge representation, learning systems, modelling, secondary education

Abstract The authors introduce a didactic system for object-oriented modelling (OOM) as a new and flexible learning concept for informatics education and outline a development methodology which is in evaluation. With the aim to improve the quality and individualization of lessons the following components of a didactic system have been identified so far: and-or-graphs for the representation of the knowledge structures within the subjects of OOM, classes of new exercises, which allow to apply and consolidate different modelling techniques and interactive visualisation modules for selected contents in order to support the initial contact with the contents and to illustrate underlying principles.

1. MOTIVATION

‘At present, the educational aims are completely changing from programming of small imperative solutions to modelling, construction and deconstruction of complex and object oriented systems of informatics. But there is a big gap between the didactical needs of the subject informatics at schools and the finished research results in this field. (Magenheim and Schubert 2000). This new orientation has essentially been influenced by Schwill and Hubwieser, which described a well-founded suggestion for informatics education based on fundamental ideas and modelling techniques (Schwill 1997, Hubwieser and Broy 1999). ‘Informatics modelling opens up

a general educational method of access to the understanding of informatics systems.' (Schubert 2000), (see Figure 1).

Figure 1. Informatics models in general education

'And a new Informatics Literacy, based on Discipline Integrated Informatics, will emerge.' (van Weert 1995). Independent of this conceptional change the object-oriented attempt has steadily gained importance in informatics education. In 1995 Schwill gave reasons for starting informatics education with the object-oriented approach: informatic-oriented demands for up-to-date education with powerful concepts, like extendability, encapsulation, evolutionary software development and others can be fulfilled and the didactic principle of continuation according to a spiral curriculum can be used. He emphasized furthermore that the object-oriented view gets very close to the natural manner as people are aware of their environment. Today object-oriented programming (OOP) has been established in German informatics teaching curricula, but the conceptional change has reached only a few informatics lessons yet. It is often still the implementation, now in an object-oriented packing, that dominates the lessons and the value of object-oriented modelling (OOM) does not become clear enough.

The great popularity of the object-oriented programming language Java and the learners' motivation and interest gained from its wide use in distributed systems also intensifies this problem. A flexible concept, supported by informatics modules, for learning and teaching OOM in

informatics lessons, which supports experimenting with and exploring of object-oriented concepts while forcing back the dominance of programming languages in opening new learning by doing oriented possibilities, is missing.

2. RESULTS OF RESEARCH

The approach of the didactic system for OOM is developed in the following. A didactic system is understood here as the mapping of a didactic concept to a learning supporting series of informatics modules with recommendations of application in informatics education. These informatics modules can either be exercises or so called exploration modules (see below). It is the aim of the concept to support learners and teachers in studying and teaching the basics of object-oriented modelling and also to build up a bridge between traditional and computer supported learning and teaching or, more precisely, informatics system supported learning and teaching. At least the following components belong to a didactic system for OOM.

2.1 Structuring of subject contents

Technical terms, methods and concepts of OOM are presented as nodes of a directed graph. The edges between these nodes represent an 'is-required'– or 'is-helpful-for'–relation, that structure the contents for learners and teachers in order to clarify, which contents are needed or helpful as pre-knowledge for other contents. The purpose is primarily the clear and compact illustration of how the subject contents are based on each other and also to give a brief view of the order in which contents should be studied or where can be continued due to a given pre-knowledge. Moreover it shall be clarified, which components from algorithms and data structures are inalienable for OOM and where they should be included within a course 'OOM from the beginning'. For learners this structuring can be useful for the organization of individual study phases or for the repetition of lessons. Summing up this structuring shall work as a tool in a scientific-oriented teaching and learning process. Within the design process of the didactic system the structuring will also be used for constructing a series of so called exploration modules (see below) for the object-oriented basics. This series of modules shall also reflect the structuring. In didactic publications concept maps are often used for the structuring of knowledge and for the

representation of multiple relations between concepts, but they do not make any boolish combination of relation couples possible, though. This is required here to be able to express that of a choice of contents either only one or all can be needed as pre-knowledge for a higher content. And-or-trees, which have their origin in the discussion of optimal strategies in game theory, contain such combinations already as integral components. For the given purpose a tree structure is impractical since different contents often require the same entry knowledge. To avoid repeating sub trees the and-or-tree is extended to an acyclic and-or-graph. The and-or-graph has successfully been used by the authors to structure contents of OOM, e. g. see Figure 2.

Figure 2. Example of an and-or-graph with OOM-content

2.2 Hierarchy of exercise classes

An essential constituent of a concept for learning and teaching OOM are exercise examples which allow to use and consolidate different modelling techniques. Within the teaching and learning process new contents are worked out on problems from the learners' reality to bring about a motivating learning process for them. Advanced exercises must contribute to various learning objectives, such as saving well known contents as well as acquiring new ones. With the aim to provide a construction methodology for exercises to OOM already published exercises (e.g. Rumbaugh et al. 1991) have been analysed. The transformation and selection of examples, which

have not been constructed for informatics education at school but for higher education, need the following quality criteria (Brinda 2000):

- *Emphasis of the modelling*: It is the aim of the examples, that learners get to know and use different modelling techniques, in order to be able to structure and formally describe reality sections and to transform these models iterative-incrementally in a form processible on a computer.
- *Language independence*: Only examples are selected, that avoid focussing on syntax details of a certain modelling or programming language and that stress the underlying concepts instead.
- *Complexity*: For the teaching and learning process exercises of various size are needed, beginning with small tasks to practice a single modelling step and ending at complex and multilayered modelling tasks. This complete range must be covered. The size of the exercises is restricted though. The complexity of the examples must follow the intellectual abilities of the learners. According to the given learning objectives the complexity of an exercise is limited. If it is the aim to develop certain new strategies, an adequate complexity of the exercises is required for increasing of motivation status and for developing the learner's skills (Vygotsky 1978).

If the selected concrete exercises are condensed to abstract exercise framework classes by separating them from their example context, they can be ordered hierarchically in accordance with the learning objectives and later be easily selected and combined for a concrete informatics lesson. Using the criteria mentioned above the exercise classes shown in Figure 3 have been identified for constructing a static object-oriented model.

Figure 3. Hierarchy of exercise classes for constructing a static object-oriented model

Besides the hierarchy a catalogue of example contexts must be built up to instantiate exercises from the abstract classes. Especially the following criteria qualify a suitable context:

- *Suitability for OOM*: An object-oriented procedure should be suggested by the example context and not been enforced by the teacher.
- *Experience reference*: For didactic reasons learners should know an example context with its various connections from their daily own experience.
- *Easy changeability and extendability*: Structuring techniques, like class hierarchies, abstract classes etc. show their quality only, if existing structures are changed or enlarged. So an open example context is required, in which expansions and modifications of the model structure are possible.

With the exercise classes hierarchy and the catalogue of example contexts a systematic problem-oriented entry to OOM will be opened.

2.3 Exploration modules

At first new contents shall be discovered individually and explored in a learning by doing oriented manner. Therefore interactive exploration modules must be provided for selected contents in order to support the initial contact with the contents and also to illustrate underlying principles. The question of the architecture and the offered functionality of such modules is of great importance here. The following criteria are of great meaning for the construction of an architecture concept for exploration modules:

- *Size*: It is the purpose of every single module to experiment with some few concepts of OOM in a ‘learning by doing’ oriented manner. The main advantage of a series of small modules is their flexible and selective usability. Moreover small modules can be handled with less training.
- *Combinability*: The architecture of the modules shall be constructed in a way that data interchange is made possible between the different modules.
- *Relation to an example context*: From this aspect two different kinds of modules can be distinguished.
 1. *Modules which are bound to the semantics of a special example context*: Changes in the model are shown in a separate window, that illustrates the changes in the semantics of the underlying example context. The main disadvantage is that one can only experiment with models from one example context.

2. *Modules which are independent of the semantics of a special example context:* These can be used for arbitrary models but they are less illustrative.

With the concept of the exploration modules a learning by doing oriented approach will be established for contents, which have often been worked out in the course of the programming.

2.4 Concept of application

The concept of application describes, how the components mentioned before are linked together in the teaching and learning process in a productive way. The main difficulty is, that learners must acquire knowledge in object-oriented basic principles, object-oriented system design and an object-oriented programming language at the same time to be able to individually construct object-oriented models. It is the aim of the concept to work out concepts and techniques of OOM with the learners systematically. The structuring of the subject contents gives well-founded suggestions about the order of the acquirements. The traversation of the content graph results in a possible structuring for a series of lessons. At first new object-oriented basic principles shall be studied experimentally with exploration modules. At the beginning of the teaching and learning process such modules can be of great value because learners do not have any own design competence at this stage of their informatics education. Linked with this phase a series of exercises with growing complexity can be used for practising and consolidating contents learned before. The hierarchy of the exercise classes promotes the structuring of exercise phases according to content. While the exercises serve to use and to deepen contents, the exploration modules shall help learners to acquire new knowledge and to build up hypotheses on the contents. By working with the modules and exercises knowledge and abilities in software development methods and an object-oriented programming language will be built up.

3. DEVELOPMENT OF A DIDACTIC SYSTEM

For the design of complex software systems design patterns (Gamma 1995) have been established as a successful tool since about 1995. In the context of software technology design patterns describe “cooperating objects and classes which are customized to solve a general design problems in a certain context”. Design patterns represent template-like solutions for

abstract design problems which are characterized by the four basic constituents pattern name, problem, solution and consequence section. Since the design of a didactic system is also a difficult and extensive task, it shall be tried to identify patterns for the design of didactic systems, to simplify and to make possible the construction of further didactic systems. It particularly is not the aim to force creative processes of lesson design into rigid schemes here. However, design patterns can contribute to a definition of quality standards in the Didactics of Informatics and can also simplify the comparability of informatics lessons.

With help of patterns the structuring of subject contents and the construction of exercises can be simplified. The idea shall be illustrated by a simple pattern. The similarity pattern can be used, if a new content shows a structural and / or semantic similarity with already known contents. This can be the case, if a concept of the (real) world is transferred to informatics, as it the case e. g. with the description of objects in the real world and in object-oriented models. The similar concept also can originate another, necessarily known, science area. Even a reference to a similar concept within informatics is possible (e. g. abstract data type and class). Given a structural similarity of the concepts involved the elements of the known concept form as entry knowledge the prerequisites for the corresponding elements of the new concept. A reference to known concepts can make learning easier, if the complete group of learners has the required pre-knowledge. Such a procedure is not suitable if the known concept is not familiar to the complete studying group. Moreover it can be possible, that certain ideas of the basic concept can be cognitive barriers for the new concept.

The idea of the design patterns can also be illustrated at the exercise examples. Here one can find the correspondences between the terminology of design patterns and didactics shown in Table 1.

Table 1. Correspondences between the terminology of design patterns and didactics

Design patterns	Didactics
Problem section	Learning objective(s)
Solution section	Abstract exercise(s), recommendation(s) for example context(s)
Consequence section	Reaching of learning objective(s) and increasing of competences or not

For the identification of design patterns for didactic systems sources from different areas must be consulted:

- *Technical sources:* Different authors have their own view on a science part, they structure and justify it correspondingly. Technical sources on OOM concentrate in general on an object-oriented programming

language or on analysis and design. To structure contents of OOM for informatics lessons it is necessary to also take into account components from algorithms and data structures (variable concept, parameter concept, algorithms etc.).

- *Didactics of Informatics sources*: Theoretical concepts, lesson series and exercise examples already consider the specific target group, give well-founded suggestions which contents should be taken into account and which not. It is often the problem of these sources in OOM, that for secondary education they are not age adequate because of a too low cognitive level, and that they do not correspond to the educational principle of scientificity since the meaning of the modelling remains unclear.
- *General didactic sources*: General didactic sources suggest certain structurings of learning and teaching processes, others are indicated as avoidable.

If one abstracts these representations, then design patterns for the construction of above described components of a didactic system can be identified.

4. STRATEGY OF EVALUATION

The theoretical concept was empirically examined in discussions with school heads and teachers, performance tests, by interviewing pupils and the video recording of lessons. In 1998/99 and 1999/2000 lessons at school were visited for the exploration and description of the traditional informatics lesson with main focus on OOM. Parallely a research report on analysis of teaching and learning concepts for OOM was worked out in the winter term 1999/2000. So the necessity of a concept for didactic systems was established. The theoretical conception for didactic systems was carried out under application and further development of technical and didactic foundations in the summer term 2000. On the basis of this concept there will be a first implementation in the study year 2000/2001. The concept will be enriched with recommendations for education and with teaching and learning materials. Informatics modules are designed and implemented. A process accompanying evaluation takes place in the field of teacher education with first tests of application in informatics lessons in order to receive a design feedback. In the school year 2001/2002 the main emphasis is on the qualitative and quantitative evaluation of the concept and the design methodology of a didactic system with informatics teachers. The results flow into the revision and improvement of the didactic system for OOM.

REFERENCES

- Brinda, T. (2000) Sammlung und Strukturierung von Übungsaufgaben zum objektorientierten Modellieren im Informatikunterricht. *LOGIN*, 5, pp. 39-49.
- Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1995) *Design Patterns*. Addison-Wesley Longman, New York.
- Hubwieser, P. and Broy, M. (1999) Educating Surfers or Craftsmen: Introducing an ICT Curriculum for the 21st Century. ComNEd – Communication and Networking in Education in a Networked Society. In *Proceedings of the IFIP Open Conference*, Aulanko – Hämeenlinna, June, pp. 163-170.
- Magenheim, J. and Schubert, S. (2000) Evaluation of teacher education in informatics. 16th World Computer Congress 2000. In *Proceedings of Conference on Educational Uses of Information and Communication Technologies*, D. Benzie and D. Passey (eds.), Beijing, August, pp. 181-184.
- Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W. (1991) *Object-Oriented Modeling and Design*. Prentice-Hall, New York.
- Schubert, S. (2000) The impact of modelling in informatics education on collaborative learning with school Intranets. In *The Bookmark to the School of the Future*, P. Hogenbirk, and H. G. Taylor (eds.), Kluwer Academic Publishers, Boston, pp. 247-258.
- Schwill, A. (1997) Computer Science Education based on Fundamental Ideas. In *Information Technology. Supporting change through teacher education*, D. Passey and B. Samways (eds.), Chapman & Hall, London, pp. 285-291.
- van Weert, T. (1995) Integration of informatics into education. In *Integrating Information Technology into Education*, D. Tinsley and D. Watson (eds.), Chapman & Hall, London, pp. 101-110.
- Vygotsky, L. S. (1978) *Mind in Society*. Harvard University Press, London.

BIOGRAPHY

Torsten Brinda received his diploma in computer science from the University of Dortmund (Germany) in 1998. In 1998 he became assistant professor in didactics of informatics at the University of Dortmund. From 1999 to 2001 he was manager of a multimedia project in teacher education. His research interests are concepts for learning and teaching object-oriented modelling in secondary and higher education.

Sigrid Schubert received her diploma in physics from the University of Dresden (Germany) in 1972 and her doctoral degree in didactics of informatics from the University of Chemnitz in 1988. Since 1972 she has been teaching informatics in secondary and vocational education. In 1998 she became professor in didactics of informatics at the University of Dortmund. Her research interests are: informatics at secondary schools, collaborative learning, multimedia evaluation in teacher education. She is chairman of the Department on Informatics Education of the German Informatics society (GI).