Classroom Practice Project “UNIS – Understanding of Informatics Systems”

Peer Stechert, Sigrid Schubert, Christian Kollee
Universität Siegen, Didactics of Informatics and e-Learning, Germany
{peer.stechert|sigrid.schubert|christian.kollee}@uni-siegen.de

Abstract: The paper gives a résumé of the classroom practice project UNIS with the aim to foster understanding of informatics systems in upper secondary education. To meet the demands of modern informatics education at secondary level we established a partnership between school and university. During a series of negotiations we defined competencies and described an educational model. Observations of the researchers in the classroom disclosed the development of competences during the learning process. The qualitative results are complemented by quantitative results gained through tests.

Keywords: Classroom Teaching/Practice, Secondary Education, Learner-Centred Learning

1. Motivation and Aims

Motivated by lack of learner centred experiments this research group “Didactics of Informatics and e-Learning” run by Sigrid Schubert has a strong focus on understanding of informatics systems in upper secondary education since 2005. In his PhD thesis (will be published in 2009) Peer Stechert developed an educational model and lessons to improve the theory of Didactics of Informatics in this particular field. The classroom practice project UNIS (2008-2009) was established to connect the predominantly theoretical research results (Stechert, 2006, 2008), (Stechert and Schubert, 2007) with an interdisciplinary empirically research project “MoKoM – Measurement Procedure for Informatics in Secondary Education” which is promoted by the German Research Foundation (2008-2010) (see http://www.die.informatik.uni-siegen.de/MoKoM/engl). Christian Kollee started working towards his PhD with MoKoM. This article focuses on the results of the classroom practice project UNIS which was designed in partnership between school and university.

Understanding of informatics systems comprises cognitive approaches to 1)
behaviour, 2) internal structure, and 3) implementation aspects of informatics systems. To distinguish developing from understanding of informatics systems, we focus on behaviour and internal structure of systems. Analysis of the research field as a normative approach has shown the potential of structure models of informatics systems in the learning process (Stechert and Schubert, 2007). Therefore, we focus on fundamental ideas of informatics (Schwill, 1997). Object-oriented design patterns have been applied as knowledge representations of networked fundamental ideas to foster understanding of systems (Stechert, 2006), (Stechert, 2007), (Freischlad and Schubert, 2006), (Stechert, 2008). In particular, the combination of views is potential, i.e., exercises combining views on behaviour and internal structure. According to the educational model of Stechert the approach to understanding of informatics systems is not necessarily limited to upper secondary education. Results have been presented at international conferences, accordingly, and show feasibility and acceptance of both, teachers and students. In the classroom practice project UNIS our aims are as follows:

1. Definition of competences: Students should develop competences for understanding of informatics systems,
2. Observation in the learning process: The concrete learning process within the learner-centred approach should disclose cognitive barriers,
3. Conclusion: The educational model comprises a collection of descriptions of learning objectives, student activities, exercises, experiences with students in the learning process, and lessons. Qualitative results demand quantitative approaches.

While the informatics-specific taxonomy of learning objectives developed by Fuller et al. (Fuller et al., 2007) only focuses on higher education the project UNIS contributes to a taxonomy for upper secondary education.

2. Project Design and Focal Points

In the following, we describe the UNIS project design. The starting point of the project was the demand of students: Modelling and implementation dominate current informatics education. To foster competences enabling students to cope with systems in their daily life, emphasis has to be put on understanding of informatics systems. We apply the educational model for understanding of informatics systems (Stechert, 2008) and decided to start a classroom project together with teachers. As a prerequisite of further research, in the classroom practice project UNIS we focus on curricula intervention, where evaluation is based on classroom experience and observation. It comprises a definition of competences, a description of the learning process and learning materials.

A partnership between school and university has been established in the PhD project of Peer Stechert. For UNIS this partnership continued and there have been intensive preparations since spring 2008. Three teachers of informatics at the co-
operating school have been involved in the process. One of them is very experienced in teacher training, and he is often the representative of the ministry of education in exams given jointly by university and government. A second teacher has already supervised a classroom project for understanding of informatics systems using design patterns. The third teacher supervised a second classroom project on understanding of informatics systems. She has been a student teacher at the University of Siegen and was giving the lessons in the classroom practice project UNIS. Thus, there is a team of motivated teachers that wants to improve informatics education together with researchers. In the cooperation we defined competences, discuss the general conditions of the classroom project and the contribution of the researchers during the observation phase. After a series of negotiations, we decided to conduct UNIS in an informatics course at upper secondary level in autumn 2008. There were 12 students at age 17. The students’ previous informatics knowledge had comprised object-oriented modelling and programming, class diagrams, and list data structure. Together with the teachers the competences c1-c5 have been defined including a graduation: The students

c1) develop the disposition and capability to change from using black box models to structure models,
c2) apply the von Neumann model to describe the behaviour of a system,
c3) apply models of layers to describe the behaviour of a system,
c4) transfer knowledge from a static model to a dynamic model perspective,
c5) transfer knowledge from models of layers to the von Neumann block diagram as a change of perspective.

To prove the feasibility student teachers developed exercises. The material has been evaluated, re-designed and assigned to the competences c1-c5 during several meetings of our partnership. It became obvious, that there was a discrepancy between the expectations of teachers and researchers concerning the choice of a depth-first or breadth-first approach. Therefore, we had to find a solution, which was acceptable for both parties with the students in mind. Thus, we decided to deepen models of layers and the von Neumann block diagram (Schubert, 2005). Since the 1940s the von Neumann architecture is a concept to describe structure and functioning of an informatics system and their universality. Such universality is a cognitive barrier to understanding of informatics systems, since mechanical machines only have one predefined purpose. Furthermore, students should be able to assign their activities to different levels of an informatics system like hardware, systems software, and application software (Tanenbaum, 2005). Table 1 shows the result of that joint work. According to the structure of UNIS, the researchers focused on the following aspects during the lessons:

1. questions of students,
2. contribution of student activities,
3. combination of visible behaviour and internal structure of systems.

The team of observers consisted of two researchers and two student teachers. The analysis of UNIS will be part of the final thesis of these two student teachers. To
Table 1. Student activities according to structure models of systems

<table>
<thead>
<tr>
<th>structure model</th>
<th>knowledge element</th>
<th>student activity</th>
</tr>
</thead>
</table>
| 1 von Neumann architecture (VNA) | central processing unit | - individual student presentations (historical development)  
- role play |
| 2 | assembler and problem-oriented programming language | - modification of a program addressing single memory cells |
| 3 | memory and input / output | - observation and assignment of elements during the application of software to the VNA |
| 4 | processing | - description of dynamic behaviour during the application of software by sequence diagrams |
| 5 computer model of layers | connection between hardware and software | - open an older discarded hard drive for demonstration purposes |
| 6 | system and application software | - group presentations of historical development  
- role play |
| 7 TCP/IP model of layers | protocols | - examining protocol layers using an interactive visualisation of data exchange |
| 8 | way of information | - describing the protocols in a model of layers |

Complement these qualitative results, there are two tests, one at the end of every structure model. Qualitative results permit developing psychological instruments to measure competences quantitatively. In a next step, a refined classroom project will focus on quantitative means analysing a representative population.

### 3. Classroom Experience and Observations

Students construct their knowledge and need to prove their cognitive model by matching different views on informatics systems. Exercises have to bring together these views to foster understanding of behaviour and internal structure of informatics systems. Behaviour of an informatics system can be investigated by observation and in informatics experiments. Hypotheses have to be proved in experiments. An example exercise illustrates how exercises can foster understanding of informatics system: Students should describe what happens when a program is launched. Therefore, they have to learn that the Operating System manages the input and activities within the Graphical User Interface and sends it to the Central Processing Unit (CPU). These components belong to a model of layers, i.e., a structure model of informatics systems. Furthermore, a sequence diagram enables
the students to visualise dynamic aspects, i.e., the behaviour of a system.

Figure 1 shows a simple diagram describing the launch of a program. It is possible to refine it even more by adding additional components to present the activities within an informatics system in more detail. In the following, we describe the development of students’ competences (c1-c5) and the corresponding learning process including student activities.

**c1) From black box models to structure models**

The UNIS project started with student talks about first computers e.g., the Abacus, Leibniz’ digital mechanical calculator, Babbage’s Analytical Engine, and the computer architecture developed by John von Neumann in the 1940s. The students discussed the functionality of typical machines. They have identified increasing automation and mechanisation, which have been combined in modern computers. Therefore, the students used the Input Processing Output principle (IPO), which describes the behaviour of an informatics system as a black box. Afterwards the von Neumann block diagram has been developed in class. It describes the structure of most modern informatics systems. However, there was a cognitive barrier matching the block diagram of the von Neumann architecture to a real machine. Almost all students participating in UNIS did not have any experience with the hardware of a typical computer and could not connect the different model elements to the corresponding hardware components. Without this imagination the abstraction level between the block diagram of the von Neumann architecture and the machine in front of them was too high to be understood.

Another cognitive barrier in such field unknown to most students is use of terminology addressing hardware appropriate for upper secondary level. The students had to identify, define, and appropriately use the technical terms associated with an informatics system (ACM, 2003). An example was the term “memory”. The students used it for different components of hardware. In many situations they thought of “main memory”, while in some cases they addressed a hard disk.

**c2) Apply the von Neumann model to describe the behaviour of a system**

In a next step, the students should understand two key aspects of the von Neu-
mann architecture by analysing program execution. At first the differences between program data (e.g., files forming an application software) and user data (e.g., “.txt” or “.mp3” files), which reside in the same memory, are crucial, because they form the basis of universality of informatics systems.

The second aspect is the division between input and output devices. To understand both, the students should familiarise themselves with three small programs.

Figure 2. Program and user data in volatile and non-volatile memory

Guided by central questions the students explored the different steps in the program flow, e.g., program start, and loading the program data from the hard drive. They also identified the components, e.g., input, output and the control unit of a von Neumann computer, which were used during the program execution according to the previously discovered steps. With the help of the von Neumann model, the students could imagine how the informatics system behaves, e.g. the execution of a program requires to load the program data from an external drive into main memory, executes it on the CPU using data provided by the user (loaded from the hard disk or via input devices), displaying and saving the results (Figure 2).

After this exercise most of the students have understood the existence of program and user data, but the fact that program data resides in main memory was not easily understood, even if it has influence on the performance of the computer. Large programs occupying main memory imply the need to access the hard disk more often, which takes more time. A possible reason for these problems could be the missing emphasis on dynamic aspects by the model. The students know the components and their behaviour, but the chronological ordering of some actions was not clear, therefore a sequence diagram, which describes the dynamic aspects, was used to visualise the cooperation of the components (Figure 1). A main characteristic of a universal informatics system is that it is independent of a concrete software running. This is realised by the so called fetch-decode-execute cycle that describes the internal operations of a CPU. At first a command is fetched from memory, decoded by the control unit into basic instructions and executed by the arithmetic-logic unit (ALU). To understand this cycle, the students were divided into two groups. The first group was introduced to a role playing game. During this game the student’s simulated parts of the CPU and interacted according to the defined rules. These rules denied any forms of conversation except the exchange of previous prepared notes. The second group of students used a Java applet, which simulates the fetch-decode-execute cycle. They received a worksheet with
additional information and some explanations about the simulation applet. In the follow-up the groups were mixed pair wise and the students should share their experience. Observations by the researchers showed that the role play was helpful to understand the simulation and therefore the fetch-decode-execute cycle, because its rules used predefined terms, which helped to discuss the simulation.

**c3) Apply models of layers to describe the behaviour of a system**

To understand the internal operation of an informatics system even further, the students were introduced to a simple phone billing application. They were asked, whether a computer can calculate the bill given by the formula \( k = (a - b) \times 2 + 16 \) within one step. Knowing that a von Neumann computer is only able to calculate on at most two operands at a time, the students had to describe which steps are necessary to compute the formula. Therefore they were introduced to a reduced form of a higher programming language, which uses a three-address-code format. Such a format allows only the usage of operations with two operands in one assignment step. In the next step this was further reduced to a simple assembly language. There was a competition between the students as to who is able to use the lowest number of variables. Such competition is very helpful to increase motivation of students as a facet of competence.

Since most students use the Internet in their daily life it has been chosen as a motivating example to introduce another model of layers, the Internet Protocol Suite, which describes the communication protocols used for the Internet and other computer networks today. The students were introduced to the different layers by using an analogy describing the communication between two diplomats not speaking the same languages. Almost all students only know the highest level in such a model, e.g., the application level. With a worksheet describing the four layers they tried to construct the structure of the Internet Protocol Suite in pairs. After a short correction of the different solutions the students watched a video that explained the functionality of the Internet using an analogy: a shop represents the internet provider, offices are servers, and persons run through a building transporting letters representing data. Students were able to understand the processes running behind the scenes of calling a web page. They tried to match the explanations onto the layers of the model. They also compiled a sequence diagram from the different objects addressed in the show, e.g., the client, the provider and the server, and the interactions between them.

**c4) Transfer knowledge from a static model to a dynamic model perspective**

During the UNIS project different structural models are used to introduce the students to various concepts in informatics. Like all structural models these express the concepts in a static form omitting the chronological order of actions between different systems components. However, dynamic aspects are necessary to gain a deeper understanding of the internal structure of informatics systems. First a role play was performed, in which the students took over the jobs of different parts of the von Neumann computer (see c2). Using a tutorial to describe the roles in the
play the students were able to act like the data bus, the ALU and other components. The first attempt was not successful but on the next try many students could combine their knowledge from the von Neumann block model and the description of the roles to get the idea of the connections between the elements of a von Neumann computer. A second approach later on was the introduction of simplified sequence diagrams mentioned at the beginning of this section. These diagrams provided a great improvement to the understanding of the whole topic. The observation of the learning process showed, that these diagrams helped a lot, as the students were able to visualise the communication between the known components of the von Neumann block model. This was confirmed by many students. As a result, these sequence diagrams will be adopted earlier during the next classroom project.

4) Transfer knowledge from models of layers to the von Neumann model

To combine the model of layers and the von Neumann block model, the students should extend a given sequence diagram. This diagram consists of five objects, the standard software, the operating system, the CPU, the graphics card and the monitor as the output device (cf. Figure 1). In addition, the standard software had already received a key stroke in form of an alphanumerical character. The students completed the sequence diagram successfully and described the actions between the objects, which lead to the appropriate character being displayed on the monitor. With the knowledge of the model of layers and the von Neumann block diagram, this exercise does not yield any problems to the students.

4 Discussion of Relations between Observation and Test

Our aim is to design an educational model for competences in understanding of informatics systems, which can be transferred to international settings. Therefore, we focussed on competences students should develop for understanding of informatics systems, the observation of the learning process, and documentation of the educational model, which comprises a collection of descriptions of learning objectives, student activities, exercises, and lessons. To understand behaviour of informatics systems it is necessary to relate it to their internal structure. To analyse students’ misconceptions and cognitive barriers, we collected their solutions during the lessons. In the middle and at the end of the UNIS project, there were tests. Different misconceptions and cognitive barriers have been observed and analysed to improve upcoming classroom projects in this field. In the following major problems will be discussed and solutions provided.

The intermediate test consisted of two tasks, where the first was made up of three subtasks (A, B, C), which were all based around the von Neumann block model. In subtask (A), the students simply had to draw the five components of the block model and their interconnections. The major problems were missing links between components or a mix of different identifier for elements of the von Neumann
model. A typical example was the mix up between memory and random-access memory (see c1). The topic of subtask (B) was to remember the fetch-decode-execute cycle of the von Neumann machine and the difference between program and user data. Most students more or less confused the fetch-decode-execute cycle with the IPO model discussed some lessons before. The last subtask (C) was similar to the exercise described above, where the students had to analyze a given application for computing the phone bill (see c3). Students should describe the (hardware) components used and at least the available data before, during and after program execution. The distinction between hardware components and model elements was difficult for students. In the second task a program had to be transferred into a simple assembly language as another knowledge representation. Ignoring some minor programming mistakes, like exchanging some commands or forgetting to define one or more variables, most of the students received almost all points. However, most of the students had unnecessary load and store instructions, like saving the accumulator into memory and loading it right away.

The second test consisted of two parts. The tasks in the first part were related to the layers of the Internet Protocol Suite. Besides the reproduction of the different layers, the explanation of the word ‘protocol’ and an assignment of different protocols to the associated layer the students should develop their own model of layers given by a short story. The story was analogous to the Internet Protocol Suite and describes the communication via letter post between two business people. Most of the students solved this task very well and were also able to draw a connection from their own model to the layers of the Internet Protocol Suite. The second part of the test consisted of tasks associated with the model of layers of computers, i.e., hardware, system software, and application software. The students had to express the different layers in their own words and depict some examples for each layer. As an additional question, the students should explain how a single processor machine handles multiple tasks at the same time. This was not explicitly covered during the lectures, but most of the students were able to give an appropriate answer including some sort of process scheduling.

5. Conclusions

In this article, we have presented results of the UNIS project for understanding of informatics systems. Our aim was to design an educational model in a partnership between school and university, which can be transferred to international settings. Therefore, we focused on competences students should develop for understanding of informatics systems, the observation of the learning process, and documentation of the educational model, which comprises a collection of descriptions of learning objectives, exercises, student activities, and lessons. The students were able to understand interaction between components of an informatics system by role playing. They were able to formalize such processes within sequence dia-
grams. Observations by researchers disclosed that students had difficulties in matching elements of structure models to the corresponding hardware components. This connection has to be considered more deeply during further classroom projects. As a next step of the design, intervention, evaluation cycle of curricula development, there will be a further classroom project in autumn 2009. However, student activities fostering understanding of informatics systems can be based on structure models. Putting emphasis on everyday situations applying informatics systems the educational model can be transferred to international settings. Since the UNIS project took place in a regular course at upper secondary level where students had typical previous knowledge qualitative results permit developing psychological instruments to measure competences quantitatively. This will be done within the interdisciplinary empirically research project MoKoM.

References


