Deconstruction of Socio-technical Information Systems with Virtual Exploration Environments as a Method of Teaching Informatics

Johann S. Magenheim
Department of Computer Science
University of Paderborn, Germany
Fürstenallee 11, 33102 Paderborn
jsm@uni-paderborn.de

Abstract: The working group Didactics of Informatics at the University of Paderborn develops and evaluates a multimedia exploration platform for information systems (MEPIS) to the needs of teaching and learning informatics at secondary schools. The paper describes the basic ideas within a system-oriented approach of didactics of informatics and two of its most relevant ideas: the perception of a socio-technical information system and the concept of deconstruction. The pedagogical and technical requirements of developing multimedia tools and integrating them into teaching and learning processes in informatics will be outlined, and the demands on a multimedia-based exploration environment for the process of deconstruction will be described. Finally, a concept of evaluation of teaching and learning processes in informatics under the perspective of a system-oriented didactical approach and the practical use of an exploration platform at secondary schools will be presented.

Socio-technical information systems in a didactical perspective

By socio-technical information systems (IS) we understand the unity of software including the graphical user interface (GUI), the hardware, embedded systems for control and regulation of peripherally technical processes and for communication with other IS and, last but not least, the associated social action system of persons, who are interacting with the IS and with other people. The technical part of an information system is exceedingly connected with its social part, by human computer interaction (HCI) and further direct or indirect technical functionalities of the information system, affecting the interactions of humans with the system and the interaction between persons. The social interaction in the context of the information system extends this to a socio-technical information system. As an example, for instance, an aeroplane can be mentioned. Information systems, e.g. embedded systems, are closely linked with the surrounding technique of the airplane to a local network of hard and software. The crew acting with this information system via the GUI of the board computer and the software implemented there extend this system to a socio-technical information system. Safety of the plane and the passengers do not only depend on the correctness of the soft- and hardware but also on the ability of the crew to interact capably with this system. The
term of the socio-technical information system, in which an action system formed by interacting persons is merged is a subsystem, has its science-theoretical roots not only in computer science but also in the sociology of technique (viz. e.g. Ropohl 1999).

The software of socio-information systems represent fundamental ideas and methods of informatics and also the concept of workflow and patterns of social action within the system. Social action, social roles of people and the workflow in the system's context will be influenced, e.g. by human computer interaction (HCI) with the technical parts of the system moderated by the graphical user interface (GUI). In the process of software development a model of the future system's functionality and its integration into the working and social context has to be generated. Modelling and developing software under this perspective is a highly communicative and interactive activity and needs close cooperation between developers and customers. Therefore, we need cooperative and evolutionary concepts for software development with stepwise refinement of its functionality. The technical implementation of a computer science system can therefore be used to prove the consequences of decisions in the context of the modelling process and cooperative action between developers and customers.

Objectives of a system oriented didactical approach

An important lesson for teaching informatics is that programming - in the sense of coding - is only one part of the construction process of an IS, and that it can no longer stand alone as the centre of a curricular concept. Techniques of system modelling and of evaluation of existing information systems as well as theoretical aspects of informatics should also be a relevant subject of informatics work in the classroom. Scaffolding curricular concepts of

teaching informatics at secondary schools which focus on the modelling aspects of software development and the social dimensions of IS are important issues.
Instruction using a system oriented didactical approach in informatics should include these essential objectives:

- Teaching fundamental concepts of informatics (like algorithms, methods of software technique..)
- Learning about (computer-based) modelling techniques.
- Recognising software development and the construction of IS as a communicative and co-operative process, i.e. construction decisions and group interests should be balanced.
- Learning, that the social impact of an implemented IS has its roots in the phases of requirement definition, specification and design of software
- Creating technical systems and IS is not only a technical but also an important social process with large influence on society

Realising this approach fosters not only the knowledge of fundamental concepts of computer science but also important comprehension and orientation in an engineered world and thus substantial aspects of general education are obtained in informatics at school.

Fostering modelling abilities of students by deconstruction of IS

Beyond algorithms, small software development, or the construction of a tiny IS classroom project, informatics also needs computer-based tools for modelling and for the exploration of existing IS. In complement to the method of constructing software, the method of deconstruction of software is a methodical alternative in informatics lessons. In addition, it also gives students the opportunity for discovering styles of learning and focuses in a special way on the modelling and design process as well as on the social implications of information systems.
The term "deconstruction" comes originally from the methodical concept of philosophy and science of literature (viz. e.g. Derrida 1997). It aims to analyse the structure and content of text and to interpret the author's opinion. Also it intends to find out about things, which have not been said by the author, but nevertheless were very important for the author’s message. Deconstructivism can also be found in the fields of architecture and arts, where we may discover traditional and new concepts of design, logic, formal and informal structures, implicit messages, and so on. Why not transfer this concept in the area of teaching software techniques?

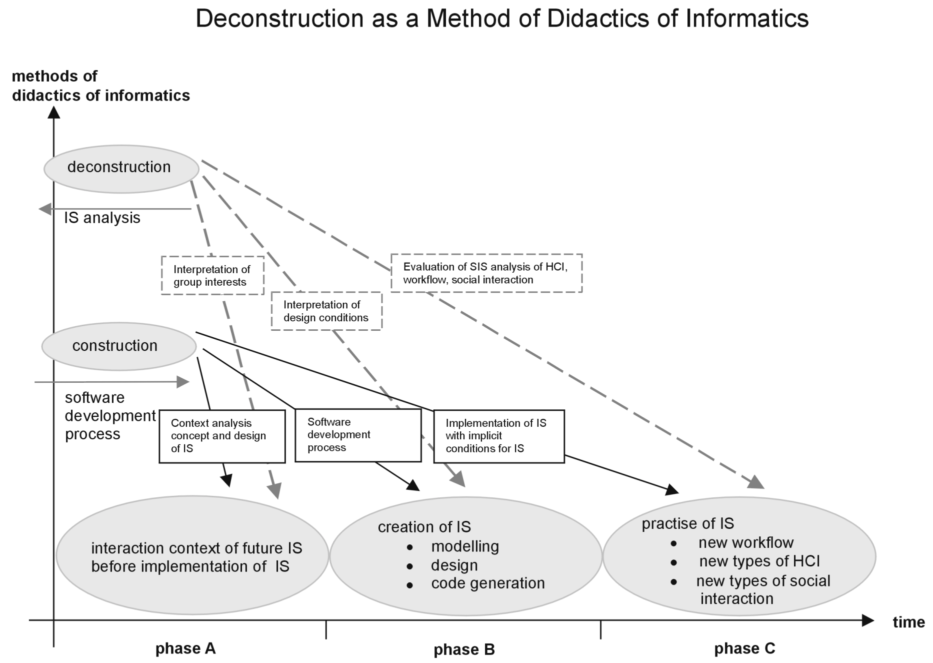
Deconstruction as a Method of Didactics of Informatics
methods of
didactics of informatics
deconstruction
IS analysis
Interpretation of group interests
Interpretation of design conditions
Evaluation of SIS analysis of HCI, workflow, social interaction
construction
software development process
Context analysis concept and design of IS
Software development process
Implementation of IS with implicit conditions for IS
interaction context of future IS before implementation of IS
creation of IS
• modelling
• design
• code generation
practise of IS
• new workflow
• new types of HCI
• new types of social interaction
time
phase A
phase B
phase C

Figure 1: Deconstruction as a method of Didactics of Informatics

Software has different forms of appearance and allows different views to discover it: We could look at the source code and see classes, methods, algorithms, programming language structures or even informatics concepts such as e.g. the problem solving method 'divide and conquer'. We might look at the GUI and learn about software ergonomics as well as about the functionality of the software. We might learn about the organisation of work within the IS and the abilities people must have to handle this product. UML diagrams (unified modelling language) and documents that have been generated during the phase of design-decision complete the arrangement. It's like a puzzle and pupils might look for the pieces and put them to the whole. The complexity of a multimedia-based exploration environment should be scalable so that it might be adapted to the level of different learning groups. When constructing software by themselves, pupils have to obey the chronology of the process and then, due to the lack of time, the software to be developed cannot be very complex. Deconstruction allows a kind of time travelling and offers a simultaneous look at different stages of the software development process, i.e. the preliminary interaction contexts of an IS before its development, the phases of software development, and the implementation and change of social context of the IS. Software design decisions may become a subject of discussions and alternative concepts of specification could be conceptualised. In addition, supplementing modules of the software could be constructed and systems' functionality could be changed. Thus, the fundamental informatics concept of re-engineering could be realised in the classroom. The main objective of this endeavour in an exploration environment is to strengthen the pupils' sensibility towards the decision-making character of modelling processes and to let them learn about the methods of modelling and software techniques. Thus informatics at school will be able to realise some essential objectives of general education: contribution to the school student's problem solving competencies, fostering social competencies e.g. (ability to work in a team, communication abilities), ability to use cognitive tools not only in informatics but also in other areas, and last but not least the ability to value the social impact of changes in technology, especially in information technology.

Virtual exploration environment MEPIS

To realise this didactical concept it is necessary to develop didactical software with open access to the source code and a multimedia-based exploration environment, which offers the different tools and types of documents such as: Java development kit, object browsing system, examples of CRC-modelling, description of use cases, description of social context of IS (patterns of interaction, types of workflow), video sequences of social action representing the use cases, interviews with future users, interviews with developers and applicants, documents concerning with the history of ST and IS, parts of documentation of the concrete IS, fragments of source code, elements of the GUI of the software, description of methods to develop, modules of software to develop, UML class and sequence diagrams, GUI prototypes, alternative software design concepts, software development strategies, scalable prototypes of software to develop and so on.

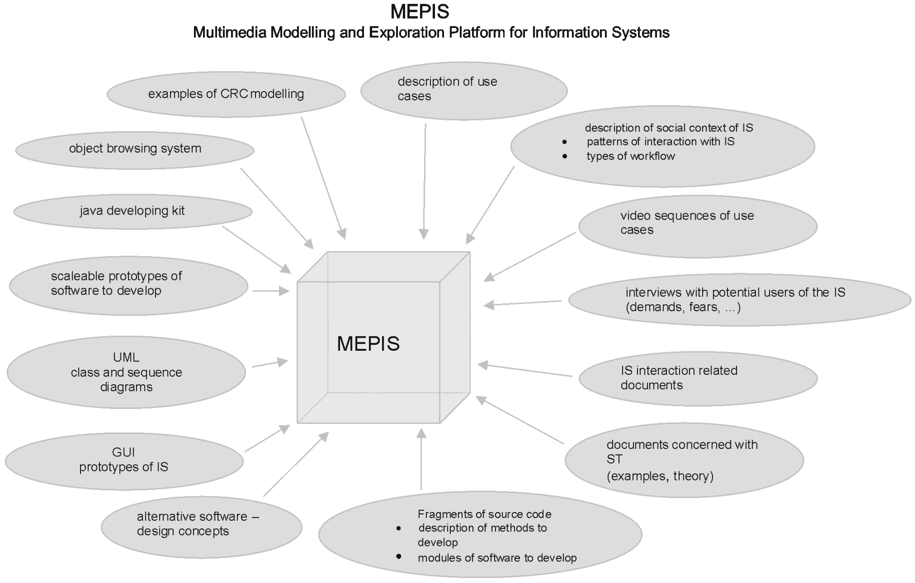
MEPIS
Multimedia Modelling and Exploration Platform for Information Systems
examples of CRC modelling
description of use cases
description of social context of IS
• patterns of interaction with IS
• types of workflow
object browsing system
java developing kit
video sequences of use cases
scaleable prototypes of software to develop
interviews with potential users of the IS (demands, fears, ...)
MEPIS
UML class and sequence diagrams
IS interaction related documents
GUI prototypes of IS
documents concerned with ST (examples, theory)
alternative software – design concepts
Fragments of source code
• description of methods to develop
• modules of software to develop

Figure 2: Components of MEPIS

The scalable exploration environment should be used for different learning scenarios in informatics at school. It should also be integrated into the teaching and learning process of the class. In an informatics project, which e.g. is to construct a small piece of software, pupils may use the exploration environment to answer questions that arise during their classroom work. The software represented in the system could be extended by creating new modules. MEPIS also represents the concept of learning by example. Informatics-related problem solving, constructing software, learning about informatics and deconstruction of an IS with MEPIS should be an integrated part of classroom work.

Questions of evaluation in a system oriented didactical approach

Learning and teaching processes are very closely linked and represent different views on the same matter. Therefore, the main subjects of empirical classroom research should be the teacher (teaching process), the students (learning process), to include the classroom interaction between students and the teacher and among the students themselves. Also, the human computer interaction (HCI) of the students with different computer-based tools and with the multimedia exploration environment (MEPIS) should be empirically evaluated.

There are some very interesting empirical studies dealing with different subjects of learning at schools. The international TIMMS-study about teaching-styles and students' conditions of learning mathematics produced especially remarkable results. But there is a lack of such empirical studies in regard with informatics at schools, and there are none regarding the special research interests of a system oriented didactical approach of teaching informatics at secondary school. Due to the perception that it is an important task of computer science to model and create information systems that have a tremendous impact on the associated social interaction system and even on the whole society, modelling and creating a small information system has always been an important classroom activity in informatics. With a system-oriented approach, a small IS, represented in a multimedia exploration environment, could be analysed and deconstructed, its socio-technical elements then bared for further learning and discussion. Empirical analysis of informatics at school under the perspective of an system oriented didactical approach pose some of the following questions:

Does the method of deconstruction and the use of MEPIS foster the students' abilities of modelling a small IS and do they acknowledge the fact that the functionality of IS depends on design decisions that could be influenced by the interests of applicants, developers, users or given economic circumstances? Does the use of deconstruction and of MEPIS change the objectives and the practice of informatics in school from a programming language course into a course in which fundamental informatics concepts are submitted? Does the method of deconstruction and the use of MEPIS open the pupils' view for the social impact of information systems and help them realise that an IS implicates an associated system of social action? Does deconstruction in terms of general education not only foster formal problem-solving abilities but also social competencies like teamwork and communicative competencies? Will deconstruction contribute to technical and general education by fostering the comprehension of the possibilities of forming a socio-technical information system in different ways? What is the contribution of a multimedia exploration environment of IS to informatics at school? Can it increase the effectiveness of computer-based modelling techniques und create a constructivistic learning environment? These are some of the essential questions of the didactical research project.

The observed contradictions between issues and results in educational practice motivates a new strategy for application of didactics of informatics. Therefore, recommendations, materials and multimedia tools have to be designed and tested. The MEPIS-project supports a new kind of relation between research, training and educational innovation. At first, future curriculum of informatics education has to be defined and a desired learning-process to be described. Thus, a number of hypotheses need an empirical evaluation in the context of the research. The courses can be videotaped at different periods to analyse case studies. Analysis focus could be the structure of the lessons, the methods used in problem solving, and the students' skills of thinking. The research group has to decide which parameter of the concept has to be changed to the next step to improve the solution. During a longer period, each step of evaluation can be manifested in a multimedia database. The possibility of storage and retrieval of large quantities of empirical data has a long tradition in didactics. What is new is the access to multimedia data and the ability to use it to analyse and compare complex learning-scenarios. Therefore, the empirical research concept must be acknowledged as a long term study.

Evaluation strategies of teaching and learning processes in informatics

Empirical data can be collected by using and combining qualitative and quantitative empirical research strategies and instruments like interviews, questionnaires, observing and videotaping lessons, assessment of course materials, or group discussions about observed teaching-practice. To obtain more empirically founded information about teaching and learning informatics it will be useful to examine the classroom interaction between pupils and teacher and among the pupils themselves. Further objects of empirical analysis of classroom practice could also be teaching-styles, organisation of classroom work by the teacher, course materials, writings on the blackboard, curriculum concepts of the lessons, source code and documentation of the software developed by the pupils, keystroke-recording of human-computer interaction situations in the classroom, results of running tests of the pupils' designed software, media support and the technical equipment in the classroom, pupil records in their exercise books, opinions of teacher and students about informatics and the objectives, the content and the practice of the attended lessons. An important source of data with didactical relevance will also be the evaluation of navigation patterns with MEPIS. Qualitative data covering the whole classroom interaction process may mainly be collected in a sample of some classes at few schools. Quantitative data will also be collected in control groups to value the influence of teaching-methods, didactical concepts of informatics and the role of multimedia-based exploration environments. For evaluation purposes and recognition of the results of empirical classroom research, it will be useful to compose clusters of students which are homogenous with respect to relevant research variables like cognitive and methodical precognition, abilities and learning types or the influence of gender.

The implicit theoretical concepts of didactics of informatics will also be empirically examined in discussions with heads of faculties or schools and the teachers by interviewing students and by the analysis of video-recording of lessons (Magenheim 1999).

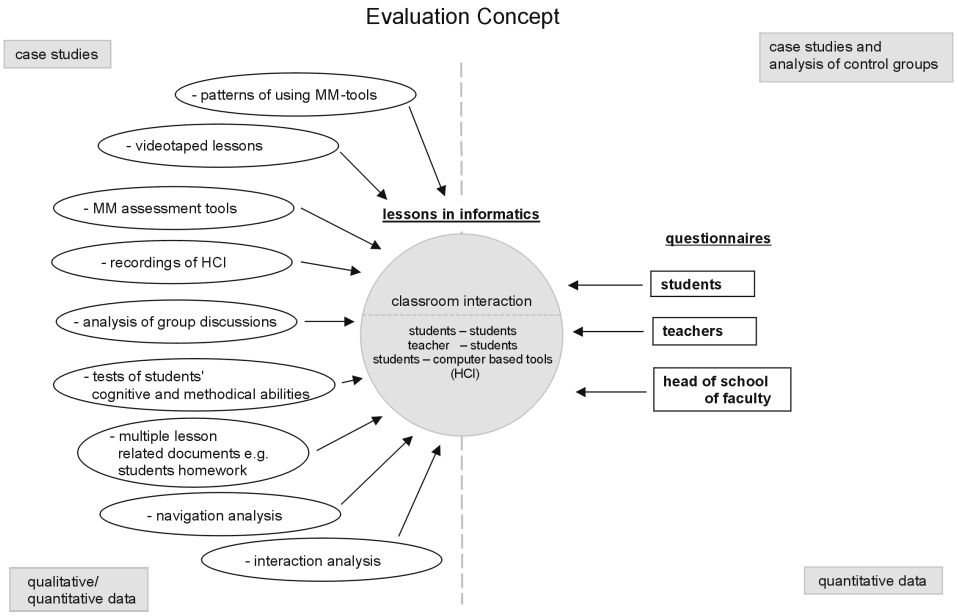
Evaluation Concept
case studies
case studies and analysis of control groups
- patterns of using MM-tools
- videotaped lessons
- MM assessment tools
- recordings of HCI
- analysis of group discussions
- tests of students' cognitive and methodical abilities
- multiple lesson related documents e.g. students homework
- navigation analysis
- interaction analysis
lessons in informatics
classroom interaction
students – students
teacher – students
students – computer based tools (HCI)
questionnaires
students
teachers
head of school of faculty
qualitative/ quantitative data
quantitative data

Figure 3: Evaluation Concept

To collect case-study data for empirical evaluation of classroom scenarios it is necessary to record a lot of classroom interactions. Therefore, classroom practice has to be observed and videotaped by two cameras, one for the teacher’s and one for the students’ perspective. Sometimes, it even might be necessary to use not only fixed but also portable cameras. The operator is allowed to walk around in the classroom to record discussions within groups or to visualise pupils' interaction with the computer. But videotaping classroom interaction for evaluation purposes has to be handled very considerately, trying not to disturb the sensitive communication context between teacher and pupils and to minimise interference from outside.

Special attention must be given to the analysis of MEPIS. We want to find out patterns of use and navigation with MEPIS. What kind of knowledge and methodological competence do the students gain by using the exploration environment and how will they transfer their competences to solve informatics-related problems at school? This may answer some questions concerning the integration of constructivistic virtual exploration platforms into a concept of traditional methods of teaching in classrooms. We will use well-known empirical instruments of human computer interaction and software ergonomics and combine them with methods of classroom research to find answers.

Conclusions

Evaluating learning- and teaching-processes is an important subject of didactics of informatics at university and has to be integrated in informatics teacher-training at university. Evaluating classroom practice and the use of multimedia exploration platforms in informatics serves different purposes.

Methodology and the tools of research, also the results of the empirical study, have to be connected with the practice of teacher education in informatics. It is very important for teacher students to learn about the educational practice of informatics at school. They will come to know about differences between theoretical concepts and practice of informatics lessons. Evaluating learning informatics is necessary not only to produce new findings or get information about current results from other research projects but also to understand empirical research methods. In a process of lifelong-learning future teachers should be able to enlarge their knowledge about teaching by studying further research results on classroom practice in informatics. Founded on a critical judgement of these research results and due to their own experience future teachers may improve their teaching-practice.

Finally, to foster these skills it is important to offer teacher students in seminars at university the opportunity to gain their own experience with classroom situations, discussing and evaluating their own behaviour in the classroom and to strengthen their faculties for self-assessment. Thus, teacher training at university will make a substantial contribution to improve teacher students' abilities for lifelong-learning and, as a result of this, in long term to increase the quality of informatics at school.

We hope that integrating multimedia tools into the their own teaching practices will not only enlarge the teacher students' methodical abilities but will also contribute to the understanding of the concept of modelling and to the pupils' comprehension that socio-information systems are formable in different ways.

As another spin-off of the MEPIS-project, the video-tapes and the variety of lesson-oriented materials may be used to establish a "best practice" data base of lessons in informatics, which may also be used to improve quality of teacher training and thus finally, the quality of informatics at school.

References

Beck, K., & Cunningham, W. (1989). A Laboratory For Teaching Object-Oriented Thinking. *OOPSLA'89 Conference Proceedings 1989, New Orleans, Louisiana. SIGPLAN Notices*, Volume 24, No 10, October 1989

Derrida, J. (1997). Die Struktur, das Zeichen und das Spiel im Diskurs der Wissenschaften vom Menschen. In Engelmann, P. (ED) *Postmoderne und Dekonstruktion: Texte französischer Philosophen der Gegenwart* (pp 114 – 139) Stuttgart: Reclam

Floyd, C. e.a.(Eds) (1992). *Software Development and Reality Construction*. Berlin e.a.: Springer

Fowler, M., & Scott, K. (1998): *UML - konzentriert*. Bonn e.a.: Addison-Wesley

Jacobson, I., e.a.(1995). *Object-Oriented Software Engineering*. Wokingham e.a.: Addison-Wesley

Magenheim, J. (1999). ViLM: Visualization of Learning and Teaching Strategies with Multimedia in Teacher Education. In *Proceedings of ED-MEDIA 99 World Conference on Educational Multimedia, Hypermedia & Telecommunications.* Seattle, WA, (pp 1593-95)

Magenheim, J., & Schubert, S. (2000). Evaluation of Teacher Education in Informatics. In Benzie, D., &Passey, D. (Eds.) Proceedings of Conference on Educational Uses of Information and Communication Technologies, 16th World Computer Congress 2000. China Beijing, (pp 181-184)

Ropohl, G. (1999). Philosophy of Socio-Technical Systems. In *Society for Philosophy and Technology* Vol. 4, No 3, URL: http://scholar.lib.vt.edu/ejournals/SPT/v4n3/

Tondl, L.(1999) .Information and Systems Dimensions of Technological Artefacts. In Society for Philosophy & Technology, Vol. 4. No. 3, URL: http://scholar.lib.vt.edu/ejournals/SPT/v4n3/

Wegner, P. (1997). Why Interaction is More Powerful Than Algorithms. In *Communications of the ACM (*pp 81-91) Vol. 40, No.5.

TIMMS, e.g.: http://nces.ed.gov/timss/TIMSS-R/more.asp