Towards a Specification of Distributed and Intelligent Web Based Training Systems

Claus Moebus, Bernd Albers, Stefan Hartmann, Heinz-Juergen Thole, Jochen Zurborg

Innovative Learning and Teaching Systems,
Department of Computing Systems,
University of Oldenburg

Keywords: Architectures, Authoring Systems, Cognitive Approaches, Distributed Learning Environments, Intelligent Distance Learning, Internet Environments, Learning Environments, Content-Engineering, Intelligent Web Based Training (I-WBT), Specification of e-Learning Systems, Unified Modeling Language (UML), Extensible Markup Language (XML), Educational Modeling Language (EML).

Abstract. Modern e-Learning Systems are expected to be innovative not only concerning comprehensive representation of content enriched by multimedia, but also in the integration of learning situations in contexts suitable for students. Suitable, motivating contexts can be "fun" as found in strategic games or business simulations or of a more "serious" variety in the form of virtual data labs. In the new BMBF Project EMILeAstat[1] (e-stat) 13 partners from different organisations are cooperating to construct such an innovative intelligent web based training (I-WBT) system for applied statistics.
This paper describes the formal specification, the architecture, and the implementation of e-stat from a knowledge and content engineering point of view, applying pedagogical and psychological criteria where necessary. Towards the end of the paper we compare our approach with an emerging e-Learning engineering approach which is based on EML a special XML-dialect.

1 The I-WBT-System "e-stat"

E-stat is an attempt to go beyond the scope of existing WBT systems by using a strong integration concept in combining well-structured content with a high diversity of methodical and didactical approaches. Special emphasis is placed on reuse and sharing of contents, clean separation of factual contents and its didactical motivated presentation, as well as the avoidance of proprietary solutions. This ambitious approach creates the need for new research and evaluation. For example, a method for the presentation of coherent and user-adaptive content

[1] The German Federal Ministry of Education and Research finances e-stat by means of the NMB funding program "Neue Medien in der Bildung" (New Media in Education).

(learning objects) supplied by a variety of sources has to be found. E-stat is motivated by pedagogical plurality. So it integrates different learning-methods, scenarios, and a consulting component into a knowledge landscape. The question is whether existing methods of specification [1] [2] [3] have to be modified accordingly, to ensure a systematic method for production of content. In the course of the project we decided to use the standard approach of software engineering modified by special educational and cognitive needs. For development and analysis purposes we make use of concept and notations supplied by object orientated analysis (OOA) and object orientated design (OOD) [4]. For the implementation we develop a special XML-dialect to give structure to the learning objects and the learning environment.

1.1 Specifications

What is the purpose of specifications? We borrow some general requirements [5], which were published for the slight different purpose of an Educational Modeling Language (EML). A specification or an EML should meet the general requirements: (1) formalization, (2) pedagogical flexibility, (3) explicitly typed learning objects, (4) completeness, (5) reproducibility, (6) personalization, (7) medium neutrality, (8) interoperability and sustainability, (9) compatibility, (10) reusability, and (11) life cycle. We will take these criteria as a frame of reference.

Specification of IPSEs In our group we started our ITS research with the development of Intelligent Problem Solving Environments [6] (IPSEs) a special type of ITS. They are instances of intelligent problem based learning systems [7]. To us they seem to be the most effective intelligent systems for enabling problem solving learning. Though they contain a comprehensive expert system or an oracle that is able to check the correctness of students' solution proposals, they lack other expensive components like teaching or student models. The curricular component in form of a teaching model is abandoned in favor of a simple sequence of task relevant problems. In place of student models individualization is achieved by the ability of the system to respond intelligently to student hypotheses. In IPSEs an expert system and the current student hypothesis are sufficient to generate adaptive help. The development is based on a cognitive meta-learning theory, which we called ISP-DL-Theory, an acronym for "Impasse-Success-Problem-Solving-Driven-Learning" [8]. This theory is influenced by the cognitive theories of Anderson [9] [10], Newell [11], and Van Lehn[12] as well as by the motivational "Rubikon" theory of Heckhausen [13] and Gollwitzer [14].
To guide the work of our group we developed an abstract specification of the IPSE philosophy. We define formally the concept of a hypothesis in a knowledge revision framework. We show that hypothesis testing can be integrated into theory revision [15] and knowledge acquisition processes of an abstract problem solver. Stating and testing of hypotheses is the most important concept in the development of IPSEs. Though most have an intuitive idea what a hypothesis is

we have to give a formal definition. We try to be as abstract as possible so that hypothesis testing in various IPSEs can be extended as special cases. The main points are summarized in Figure 1. According to ISP-DL theory there are several

(1) ***Problem Solving:***

$$S \models E$$

(2) ***Incorrect Proposal:***

$$T \neg \models E$$

(3) ***Stating Hypotheses:***

$$E = E_{fix} \cup E_{mod}$$

(4) ***Completion Proposal:***

$$T \models E' \quad \text{mit:} \quad E' = E_{fix} \cup E'_{mod}$$

with desirable but Domain dependent Monotony:

$$T \models E_{fix} \quad \text{und:} \quad T \models E'_{mod}$$

(5) ***Self Explanation:***

$$S' \models E' \quad \text{mit:} \quad S = S_{fix} \cup S_{mod}$$

$$\text{und:} \quad S' = S_{fix} \cup S'_{mod}$$

(6) ***(Inductive) Knowledge Modification:***

$$S \setminus S_{mod} \cup S'_{mod} \models E'$$

Fig. 1. Problem Solving, Hypotheses Testing, Self Explanation and Inductive Knowledge Modification in IPSEs

steps when acquiring knowledge with IPSEs. (1) Using his subjective theory S the problem solver generates evidence or an artifact E, which may be a solution proposal to a task. From the viewpoint of an ideal expert this proposal may be wrong. (2) This proposal E is submitted to the system. If the proposal is in error it cannot be explained by the system's domain theory T contained in the expert system. The learner gets an according feedback. (3) Thus the system offers the problem solver to generate a hypothesis and he may partition his proposal E into two parts E_{fix} and E_{mod}. The student has the hypothesis that E_{fix} can be embedded into a correct solution. (4) Now, the system generates with its theory T a system response to the hypothesis. E' is a system generated solution proposal, which contains E_{fix}. E'_{mod} is help information for the student which in our IPSEs is shown to the student stepwise on demand. (5) After these events (hopefully) we have some knowledge acquisition events on the learner side. According to ISPDL-theory we expect some self-explanation: the student tries to explain E' with its parts E_{fix} and E'_{mod} to himself. As a result, the learner

generates new knowledge S'_{mod}. As indicated in (6), this new knowledge gives him the opportunity to understand E'. According to (6) this is an inductive inference, because S'_{mod} can be inferred inductively from $S \setminus S_{mod} \cup \{E'\}$. The comparison of (1) with (5) results in a revised theory S'.
Though this specification was a good guide for the development of various IPSEs in our group, it was not useful for the development of I-WBT systems in a multiparty consortium with several content-providers. The specification is too abstract, even if it had been translated to UML with abstract classes T, E and an entailment relation between some classes. For instance the oracle within the IPSE was not specified in detail. In some domains you need grammars and in other domains model checkers to check the correctness of student proposals. Which of the 11 requirements are not met by the IPSE-specifications: (2), (4), (9), (10), and (11). So we had to look for other specification approaches which are suitable for a distributed development.

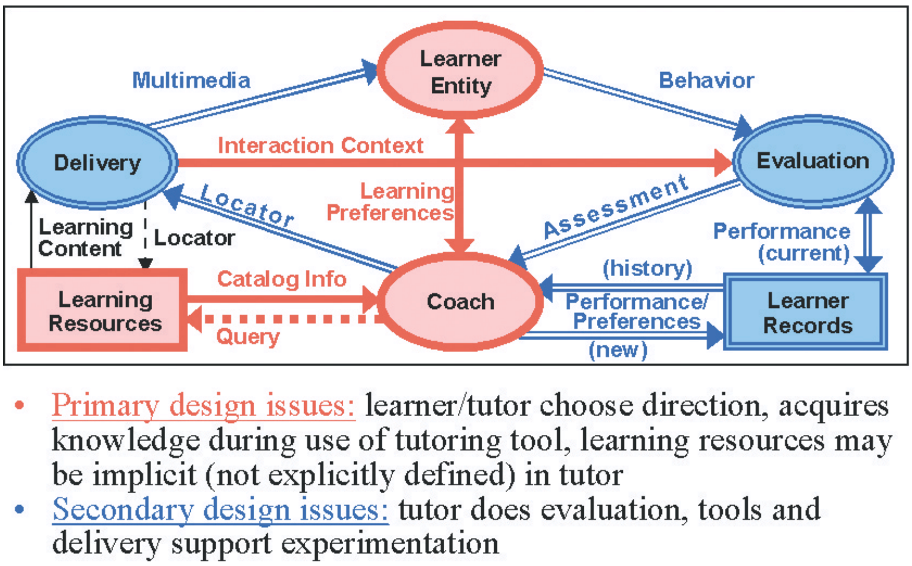

Fig. 2. LTSA mapped to intelligent tutoring tool

Specification of the LTSA Following their authors the LTSA [16] (Learning Technology Systems Architecture) specification covers a wide range of systems, commonly known as e-learning technology. The LTSA specification is pedagogically neutral, content-neutral, culturally neutral, and platform-neutral. The LTSA is neither prescriptive nor exclusive. Many systems may satisfy the requirements of the LTSA specification although they don't provide all the components,

have differing organizations, or have differing designs.
The specification (Fig. 2) mentions stores (rectangles), processes (ovals), and flows (arrows) in a kind of YOURDON-notation. Figure 2 demonstrates the view a developer should have when developing an ITS according to the LTSA-standard.
As can be seen from the Figures 1 and 2 it is possible to map the IPSE-specification to the LTSA-specification. The former is more specific than the latter. The IPSE-specification would extend the LTSA-specification if we would translate both to UML. Which of the 11 requirements are not met by the LTSA-specifications: (1), (3), (4), (6), (8), (10), and (11). The lesson learned for e-stat was, that LTSA is too vague for a distributed development in a multi-party consortium.

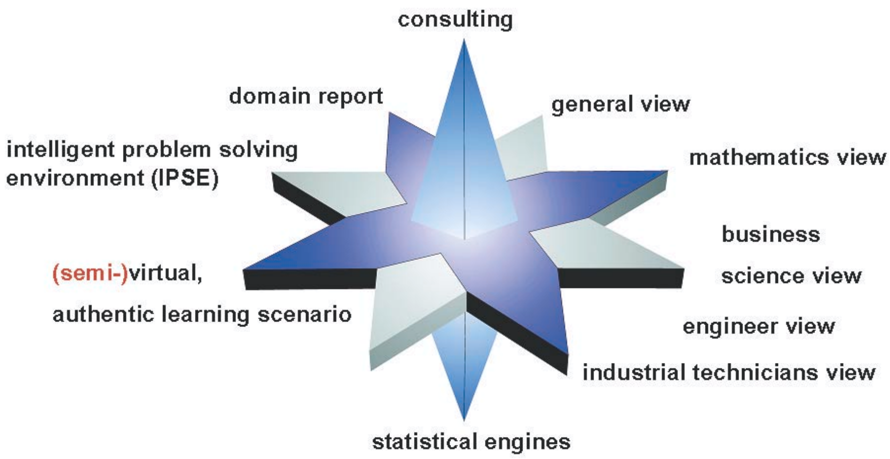

Fig. 3. The Wind Rose as a metaphor for pedagogical plurality

Specification of e-stat In the beginning of the project the wind rose [17] (Fig. 3) was used as a metaphor for the e-stat idea. It was meant to express e-stats ambition to supply applicable solutions with changing didactical demands [18] [19] (e.g. instructional, cognitive, and constructive): courses of differing levels of complexity for mathematicians, managers, psychologists and engineers but also for people with a special need of practical experience like industrial technicians. e-stat furthermore contains methods to integrate existing statistical engines, (semi-) virtual learning scenarios, an automated glossary, and the case based consulting component for the "hasty user". Next the wind rose was transferred into use-cases of the semiformal UML-Notation [20] (Fig. 4). A use-case is a typical application of e-stat. Due to the open nature of the e-stat system, the

process of defining new use-cases has not been finalized.
The next step in the OOA constitutes the construction of the static system

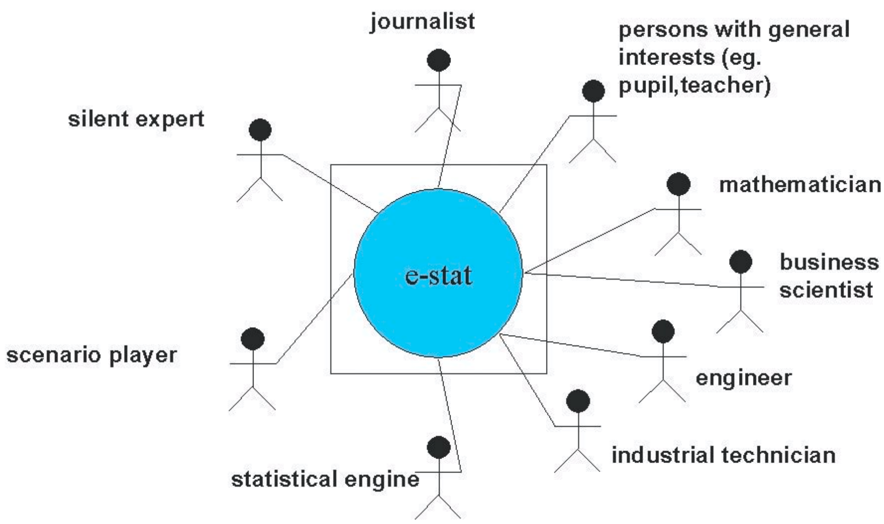

Fig. 4. Use-Cases of e-stat

structure using a class diagram. A class defines structure (attribute), behaviour (operations), and relations (associations and inheritance structures) for a collection of certain objects [21].
As can be seen in our class diagram, e-stat is a composite aggregate of views (Fig. 5). Views are shared aggregates of scenarios, courses, course units and concepts. Concepts have recursive structure. They can be built up by text blocks (text leafs) module frames, and/or concepts. This architecture ensures the representation of hierarchically organized lessons. Module frames are again composite aggregates of modules, which are the smallest building blocks or knowledge-units of e-stat.
Types (e.g. moduleType) were provided to us by the statistical content providers, which are members of the department of mathematics at our university. Inside module frames, modules are interlinked to define a partial order (e.g. "X depends on Y").
The ontological links will be specified reflexively by the association "up" inside the class "moduleframe" (Fig. 5). A conceptual map of the e-stat content can be created automatically using this pointer structure. On the right hand side of Figure 6 a cut-out of the ontology is illustrated which will be used in the consulting component of e-stat to deepen explanations on demand.
This consulting component is based on methods of case-based reasoning. The cases consist of Question-Answer-Pairs (QAPs). In the beginning the QAPs will

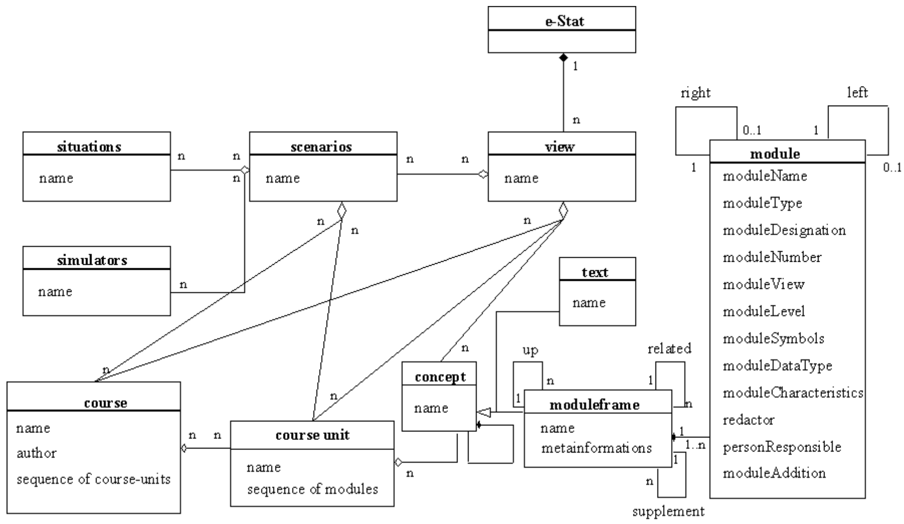

Fig. 5. Class Diagram of the Content Aspect

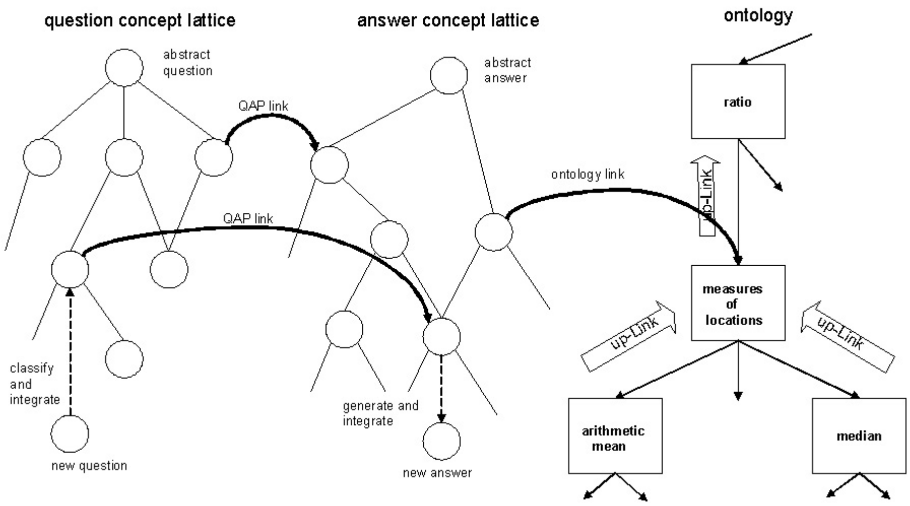

Fig. 6. The consulting component of e-stat

be extracted from consulting sessions with experts to initialize the component.
With the use of formal [22] and relational concept analysis [23] we will build up both question and answer concept lattices. The root nodes of the concept lattices are the most general question or answer node. The leaf nodes represent more specific questions respective answers.
If the hasty user asks a question, the consulting component will indicate the most similar question using the similarities given by the question concept lattice. To response the consulting component makes a search in the answer concept lattice for the appropriate answer, which is next to the question. This response is displayed to the user. If he needs detailed explanations or relevant hints for the reinforcement of his learning the ontologically structured content of e-stat serves to meet his needs.
Only if the response is not helpful for the user, the unanswered question will be transferred via asynchronous communication to human consultants. After the answer of the human expert satisfied the user, both the question and the answer are integrated into the respective concept lattice. So we have some learning mechanism in the system [24].
Due to this learning capability, the quality of the automatic consulting component will be steadily increased. After a period of time, we expect that only really difficult and interesting questions will be delivered to the consultants.
Which of the 11 requirements are not met by the momentary e-Stat-specifications: (4) and (9). The main deficits are the lack of completeness and the lack of integrating (educational) standards. The former point is not serious because the project just started 6 months ago. The latter deficit is really not a deficit, because the existing standards are not convincing.

1.2 Implementing e-stat using a 4-Tier Architecture

Most classes of the class diagram are implemented using the standardised XML language.
XML allows to create semantic tags additional to syntactic tags (HTML). Authors receive the respective document type definitions (DTDs) to generate valid class objects. DTDs for modules and module frames have been developed. Modules are specified by following attributes taken form the class-diagram (Fig. 5): moduleName, moduleType, moduleDesignation, moduleNumber, moduleView, moduleLevel, modulSymbols, moduleDataType, moduleCharacteristics, redactor, personResponsible, and moduleAddition. This module structure is the result of an interactive process between domain experts, content providers, and knowledge engineers. Depending on the separation of content and layout the authors will also get a XSL-file, which is responsible for the layout. While developing content, authors have preview permanently.
e-stat is implemented using a 4-tier architecture. The presentation tier supplies content providers and students with suitable graphical user interfaces (GUIs). The view-author GUI will be powerful enough to enable authors to construct a course for his particular target group from the certified e-stat modules by means of a system similar to "shopping cart" systems used by many e-shops. It should

only be necessary to construct new modules in very special cases.
Authors can use current XML-editors instead of a special e-stat content authoring GUI. In the logic-tier we use an Apache server, which is installed in Oldenburg. On this server our e-stat-control-system is implemented. Interactions with our database and other statistical-engines will be managed in this tier, as well as the handling of the user-administration. The native XML-database TAMINO is represented in the data-tier. Statistical engines (Xplore, SPSS, qs-stat) and scenario engines for simulation (e.g. handling business planning, production or stock exchange) are part of the application tier.

2 Related Work

During the work in our project we came across the EML initiative [25] [26].The pedagogical meta-model consists of four (conceptually) packages: (1) Theories of learning and instructions, (2) Learning Model, (3) Unit of Study Model, (4) Domain Model. These partially overlap with our approach. The parts (1) and (4) are identical to our ideas. Part (2) is a generalization of the ISP-DL-Theory. ISPDL-Theory allows more specific empirical hypotheses and more constraints for the development process of learning systems. The Unit-of-Study Model overlaps partially with the e-stat-Class-Diagram. The e-stat-Classes are at the present time partly not so semantically rich. Especially the left upper triangle concerned with scenarios has to be worked out further. Otherwise is the right lower triangle of the e-stat-Class-Diagram more semantically elaborated in comparison to the knowledge object structure of EML. What could be said at the moment is, that we could not meet the requirements of our mathematical partners using an unmodified EML. At the present moment we try to use an unmodified EML in a less demanding nonmathematical domain.

3 Contact

Prof. Dr. Claus Moebus, Innovative Learning and Teaching Systems, Department of Computing Systems, University of Oldenburg, D-26111 Oldenburg, Germany
mailto:moebus@informatik.uni-oldenburg.de

References

1. Aviation Industry CBT Committee (AICC)
2. Learning Technology Standards Committee (LTSC) mit den Work Groups IEEE P1484.1 Architecture and Reference Model WG: Learning Technology Systems Architecture (LTSA) und IEEE P1484.12 Learning Object Metadata WG: LOM Standards
3. Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE)
4. BALZERT, H., Lehrbuch der Objektmodellierung, Heidelberg: Spektrum Akademischer Verlag, 1999

5. KOPER, R., Modelling Units of Study from a Pedagogical Perspective: The Pedagogical Meta-Model behind EML, Educational Technology Expertise Centre, Open University of the Netherlands, First Draft, Version 2, 2001
6. MOEBUS, C., Towards an Epistemology of Intelligent Problem Solving Environments: The Hypothesis Testing Approach, in J. Greer (ed), Artificial Intelligence in Education, Proceedings of AI-ED 95, Charlottesville: AACE, 1995
7. BARROWS, H.S. & TAMBLYN, R.M. Problem-based learning: an approach to medical education, New York: Springer, 1980
8. MOEBUS, C., SCHROEDER, O. & THOLE, H.J., Diagnosing and Evaluating the Acquisition Process of Programming Schemata, in J.E. Greer, G. McCalla (eds), Student Modelling: The Key to Individualized Knowledge-Based Instruction, Berlin: Springer (NATO ASI Series F: Computer and Systems Sciences, Vol. 125), 1994
9. ANDERSON, J.R., Knowledge Compilation: The General Learning Mechanism. In: R.S. Michalski et. al., Machine Learning II. Kaufman, 1986
10. ANDERSON, J.R., A Theory of the Origins of Human Knowledge, Artificial Intelligence, 1989
11. NEWELL, A., Unified Theories of Cognition, Cambridge, Mass.: Harvard University Press, 1990
12. VANLEHN, K., Toward a Theory of Impasse-Driven Learning, in H. Mandl et. al.(eds), Learning Issues for Intelligent Tutoring Systems, Berlin: Springer, 1988
13. HECKHAUSEN, H., Motivation und Handeln, Heidelberg: Springer, 1989
14. GOLLWITZER, P.M., Action Phases and Mind-Sets, in: E.T. Higgins & R.M. Sorrentino (eds), Handbook of Motivation and Cognition, Vol. 2, 1990
15. DE RAEDT, L., Interactive Theory Revision, San Diego: Academic Press, 1992
16. LTSA Draft 9, LTSA Home page, http://edutool.com/ltsa/, 2001
17. Foerderantrag an das BMBF, Foerderkennzeichen 08NM058A, 2000
18. JANK, W & MEYER, H., Didaktische Modelle, Frankfurt a. M.: Cornelsen Scriptor, 1994,
19. BRUNS, B. & GAJEWSKI, P., Multimediales Lernen im Netz: Leitfaden fr Entscheider und Planer, Berlin: Springer Verlag, 1999,
20. BOOCH, G. & RUMBAUGH, J. & JACOBSON, I., The unified Modeling Language User Guide, Addison-Wesley, 1999,
21. BALZERT, H., UML kompakt, Heidelberg: Spektrum Akademischer Verlag, 2001,
22. GANTER, B. & WILLE, R., Formal Concept Analysis: mathematical foundations, Springer Verlag, 1999,
23. PRISS, U., Relational Concept Analysis: Semantic Structures in Dictionaries and Lexical Databases, Dissertation, TH-Darmstadt
24. HEINRICH, E. & MAURER, H., Active Documents: Concept, Implementation and Applications, Journal of Universal Computer Sciences 6, 2000
25. KOPER, R., Modeling Units of Study from a Pedagogical Perspective: The Pedagogical Meta-Model behind EML, Educational Technology Expertise Centre, Open University of the Netherlands, First Draft, Version 2, 2001
26. KOPER, R., From Change to Renewal: Educational Technology Foundations of Electronic Learning Environments, 2000, http://eml.ou.nl/introduction/articles.htm/