MÖBUS, C., et al., *Towards an AI-Specification of Intelligent Distributed Learning Environments* , KI - Zeitschrift Künstliche Intelligenz Heft 1/03 "Schwerpunkt:Lernen: Modellierung und Kommunikation", S. 19-24, Bremen: arendtap Verlag, 2003, ISSN 0933-1875

Towards an AI-Specification of Intelligent Distributed Learning Environments

Claus Möbus, Bernd Albers, Hilke Garbe, Stefan Hartmann, Heinz-Jürgen Thole, Vera Yakimchuk, Jochen Zurborg

This paper reflects on the possibilities to specify IDLEs (Intelligent Distributed Learning Environments). These are distributed multi-party eLearning environments with AI-components (generative expert systems, learner models, etc). IDLEs offer *intelligent, learner adapted, anytime* services to learners (eg. e-diagnosis of solution proposals, e-consulting in the case of errors). Specifications seem to become necessary to assure consistency among content providers and developers with respect to content, methods, knowledge and response spaces of learners. In this paper we want to demonstrate how to specify AI-components of IDLEs on a logical meta-level and the DLE-part of the system with UML/XML. It will be argued that the use of the new emerging eLearning specification language like the Educational Markup Language (EML) is not advisable when specifying AI-components. A more promising alternative could be software patterns or specifications in logic (eg. the situational calculus).

Introduction

While developing IDLEs in multi-party consortiums the necessity of specifications is becoming clear. Specifications serve to formalize the structure and the interaction of complex components including interrelated areas of responsibility. Specification guided design of IDLEs supports the communication between project partners, may help to avoid design errors, ensures the intended functionality, and by the way enhances the systems quality.

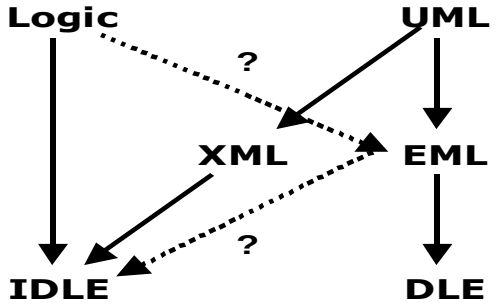

Figure 1: Influence graph of specifications

IDLEs consist of a conventional DLE and knowledge based components. In the BMBF projects EMILeAstat* („e-stat“) and mile/ET we are developing such innovative systems for applied statistics and electrical engineering. Figure 1 demonstrates the flow of ideas. At present the specification of IDLEs has two sources. The AI-part inherits its specification from machine learning[1]. We call the AI-components Intelligent Problem Solving Environments (IPSEs[2]). These are Intelligent Tutoring Systems (ITS) without student and teacher model but with a strong (generative) expert system (GXPS), which plays the role of an oracle. The GXPS is powerful enough to check user hypotheses concerning the usefulness of solution proposals. Thus adaptability is made possible without a (costly and error-prone) student model. The DLE part and the ontology of the curriculum are modeled by an UML/XML binding. The question is whether it is possible to avoid the combination of two approaches by using one eLearning specification language like e.g. EML.
The first part of this paper specifies our learner meta model: Impasse-Success-Problem Solving-Driven-Learning (ISP-DL) theory. It is sufficiently general that various accepted cognitive theories of knowledge acquisition can be subsumed. The theoretical and practical useful aspect of ISP-DL - theory is that it pinpoints learning events and the types of useful help information.

Specifications

Learner meta model: ISP-DL-Theory

ISP-DL[3] [4] [5] theory is an acronym for „Impasse - Success - Problem - Solving - Driven - Learning“. It is influenced by the cognitive theories of Anderson[6] [7], Newell[8] and Van Lehn[9] as well as by the motivational „Rubikon“-theory of Heckhausen[10] and Gollwitzer[11]. Our group use ISP-DL as a reference learner theory. It is a specialization of the EML – learner – meta model. From an AI-point of view an ISP-DL-agent is a Belief – Desire –

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

Intention (BDI) – agent[12] [13] with a deductive and inductive learning component added.
The „Rubikon"-theory describes the different problem solving phases. In the deliberate phase the problem solver considers several goals and finally chooses one as an intention. In the planning phase a solution plan is developed to compile the most prominent intention into an action sequence. Phases correspond to "mind sets" which may cause information barriers. It is important to know what type of information is helpful in certain learning phases. The plan is executed and evaluated. At impasse - time where knowledge is lacking weak heuristics are tried. If these are successful their traces are chunked to new operators.
From our own empirical investigations[14] we concluded that it is fruitful to describe learning as an interplay of impasse- and success-driven learning. Learning has two aspects: the process of knowledge optimization occurs after a solution has been found. This process is deductive in the sense that the new optimized knowledge is a logical consequence of old knowledge[15]:

old knowledge ∪ evidence |= new knowledge

The more interesting knowledge acquisition process occurs after solutions have been found with the help of heuristics to surmount impasse situations. This process is inductive:

old knowledge ∪ new knowledge |= evidence

So heuristics successful applied generate inductive knowledge. In a learner friendly IPSE hypotheses testing is a system supported heuristics.
When to expect hypotheses testing activities? We assume that the problem solver has a solution proposal for the given task. This is evaluated by mental or real time simulation or by asking an oracle (e.g. the IPSE). When there is negative feedback the student realizes an impasse. The reaction to that is replanning and using weak heuristics. One of them is testing hypotheses: that means asking the system questions concerning the solution status of parts of the original defective proposal: „Is this part of my solution proposal embeddable in a correct solution?"
ISP-DL theory motivates the following principles:
(1) The IPSE should not constrain and interrupt the problem solver but offer information only on demand. According to the theory, information is only helpful at impasse time. We think that it is important first to let the learner develop her/his own solution ideas and then later optimize his solutions. As novices are rather "creative" in generating unusual solutions the systems should be sufficiently powerful.
(2) The student should have the opportunity to obtain detailed feedback and information any time. Since impasses are possible at different phases of problem solving, the system has to offer support in all of them.
(3) The learner should be enabled to make use of her/his pre-knowledge as much as possible when asking for help. Thus the provided information should be conditioned on his hypotheses and his pre-knowledge to avoid follow-up impasses.
(4) The information provided should be tailored in grain size and amount to the knowledge state of the problem solver. If the grain size of the information is too fine or too coarse and the amount is not synchronized to the knowledge deficit, then the problem solver has to filter or generate new information, which can have undesirable emotional effects preventing progress. A student model is needed only if there is a set of help alternatives to choose from.
(5) It is necessary that the learner is free in the choice of his problem solving operators and his interaction modality. We should offer an IPSE to enable unconstrained problem solving.

Specification of IPSEs

Intelligent Problem Solving Environments (IPSEs) are designed according to the ISP-DL theory and enable learning by solving problems, testing hypotheses and self-explaining system feedback by the learner. They are instances of problem based learning systems[16] and belong to the class of constructivistic learning environments[17] [18].
We developed them in various knowledge domains of different complexity: ABSYNT (functional programming), PETRI-HELP (modeling distributed systems with petri-nets), WI\KEA (room configuration for homely care), WULPUS[19] (management game), TAT[20] (constructing constitutional formulas of reaction equations in organic chemistry), PULSE (constructing pneumatic circuits), MSAFE (constructing electric circuits), Patent-IT[21] (assistance for patent proposers).
At present the IPSEs mile/ET[22] (analyzing and simplifying electric circuits) and the LISREL-quiz of e-stat ("LI.Q.", see below) are conceptualized and implemented. Compared to ITSs IPSEs contain no teaching or student models. The curriculum is substituted by a sequence of obligatory domain relevant problems. Information is not supplied in the form of instructions but in the form of help information: pull principle. In place of student models individualization is achieved by the ability of the system to respond adaptively to student hypotheses. In IPSEs a generative expert system (GXPS) and the current student hypothesis are sufficient to generate adaptive help.
We describe briefly a typical IPSE and its formal specification. The idea of a hypotheses testing environment was first developed for ABSYNT[4] [5] [14] ("Abstract syntax trees") an IPSE for functional programming. In a diagnosis-, hypotheses- and help environment the learner may state the hypothesis that her/his solution proposal (or part of that proposal) to a programming task is correct or

at least embeddable in a correct solution. The system then analyzes the part of the solution proposal chosen by the student as a hypothesis. As a result, the system gives help and error feedback on the implementation and planning level by synthesizing correct solutions constrained by the learners' hypothesis. In principle the system can generate complete solutions but offers only help information on demand to stimulate self-explanation on the learner side.

In ABSYNT the hypotheses testing is based on diagnostic horn rules, defining a "Goals-Means-Relation" (GMR), which analyzes and synthesizes several millions of solutions proposals for 40 programming tasks. The GMR-base contains ca. 1200 diagnostic rules with fine grain size so that a large solution space, containing also unusual solutions, can be covered. However due to theoretical reasons diagnosis and solution generation in ABSYNT is correct but not complete.

To guide our work we developed a logical meta level specification of the IPSE philosophy in a knowledge revision framework. We show that hypotheses testing can be integrated into theory revision[23] and the knowledge acquisition process of an abstract problem solver. Hypotheses testing in various IPSEs can be instantiated as special cases.

(1) Problem Solving: **S |= E** or **S |- E** (subjective correct proposal)
(2.1) Possibly Incorrect Proposal: **T $\neg$|- E** **(2.2) Incorrect Proposal:** **T $\neg$|= E**
(3) Stating Hypotheses: **$E = E_{fix} \cup E_{mod}$**
(4) Sound Completion Proposal: **T |- E´** and **T |= E´** with **$E´ = E_{fix} \cup E´_{mod}$** with **desirable but domain dependent monotony:** **T |= E_{fix}** and: **T |= $E´_{mod}$**
(5) Self-Explanation: **S´ |= E´** or **S´ |- E´** with: **$S = S_{fix} \cup S_{mod}$** and: **$S´ = S_{fix} \cup S´_{mod}$**
(6) (Inductive) Knowledge Modification: **$S \setminus S_{mod} \cup S´_{mod}$ |= E´**
(7) Used Symbols: S = subjective theory (personal knowledge) S_{fix} = proven knowledge S_{mod} = modifiable knowledge T = specification of the task (=theory) E = solution proposal of the task E_{fix} = retaining part of the proposal E_{mod} = modifiable part of the proposal E´ = by the IPSE modified proposal

Table 1: IPSE meta level specification

We display the formal meta level specification in table 1. According to ISP-DL theory there are several steps when acquiring knowledge. (1) Using his subjective theory S the problem solver generates evidence or an artifact E, which may be a solution proposal to a task. The artifact E is either correct by intuition (|=) or by derivation (|-) with respect to his subjective theory. 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 wrong it cannot be explained by the system's domain theory T implemented in the expert system. The learner gets according feedback. (3) Thus the system asks the problem solver to generate a hypothesis and he may partition his proposal E into two parts E_{fix} and E_{mod}. The student proposes the hypothesis that E_{fix} can be embedded into a correct solution. (4) Now, the system generates from its theory T a response to this hypothesis. E´ is a solution proposal, which contains E_{fix} and $E´_{mod}$. $E´_{mod}$ can be used as help information for the student which is shown to the student stepwise on demand. In some non-monotonic domains (e.g. modeling with petri-nets) the parts of E may not be consequences of T. (5) After these steps (hopefully) we have some knowledge acquisition events on the learner's side. According to the ISP-DL theory we expect some sort of self-explanation: the student tries to explain the correctness of E´ with its parts E_{fix} and $E´_{mod}$ to himself. As a result, the learner generates new knowledge $S´_{mod}$. In contrast to Van Lehn's self-explanation model[24] SIERRA the correct examples are not taken from textbooks but are generated by the IPSE. As indicated in (6), this new knowledge gives him the opportunity to understand E´. According to (6) this is an inductive inference. The comparison of (1) with (5) results in a revised subjective theory S´.

Though this specification was a good guide for the development of various IPSEs in our group, it is not very useful from a software engineering point of view. 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 is not specified in detail. In some domains grammars and in other domains model checkers are needed to check the correctness of student proposals. There was a need for other specification approaches which are suited for a distributed development.

Specification of an IDLE: e-stat

In the beginning of the project we used the wind rose[25] metaphor to express the idea of a flexible learning environment for various pedagogical approaches[26] [27] [28]. To meet the demands of *p*ersonalized *c*urriculum *o*rientated *s*cenarios (PECOS-view) courses of differing levels of complexity for mathematicians, managers, psychologists, engineers, industrial technicians and even high school students are supplied.

e-stat is furthermore designed to provide *si*tuated *p*roblem *o*rientated *s*cenarios (SIPOS-view). To support this aspect e-stat provides the access to smart engines[29] (statistical engines, (semi-) virtual scenarios and a consulting engine). The system architecture is developed according the Rational

Unified Process[30]. In the first step we transferred the wind rose into use-cases. Due to the open nature of e-stat, the process of defining new use-cases has not been finalized.
In the second step the static system structure is modeled with a class diagram (Fig. 2). e-stat is a specialization of an IDLE and a composite aggregate of views. Views are shared aggregates of scenarios, courses, course units and concepts. There are two types of abstract scenarios: situated problem orientated (SIPOS) and personalized curriculum orientated (PECOS). Concepts are abstract.

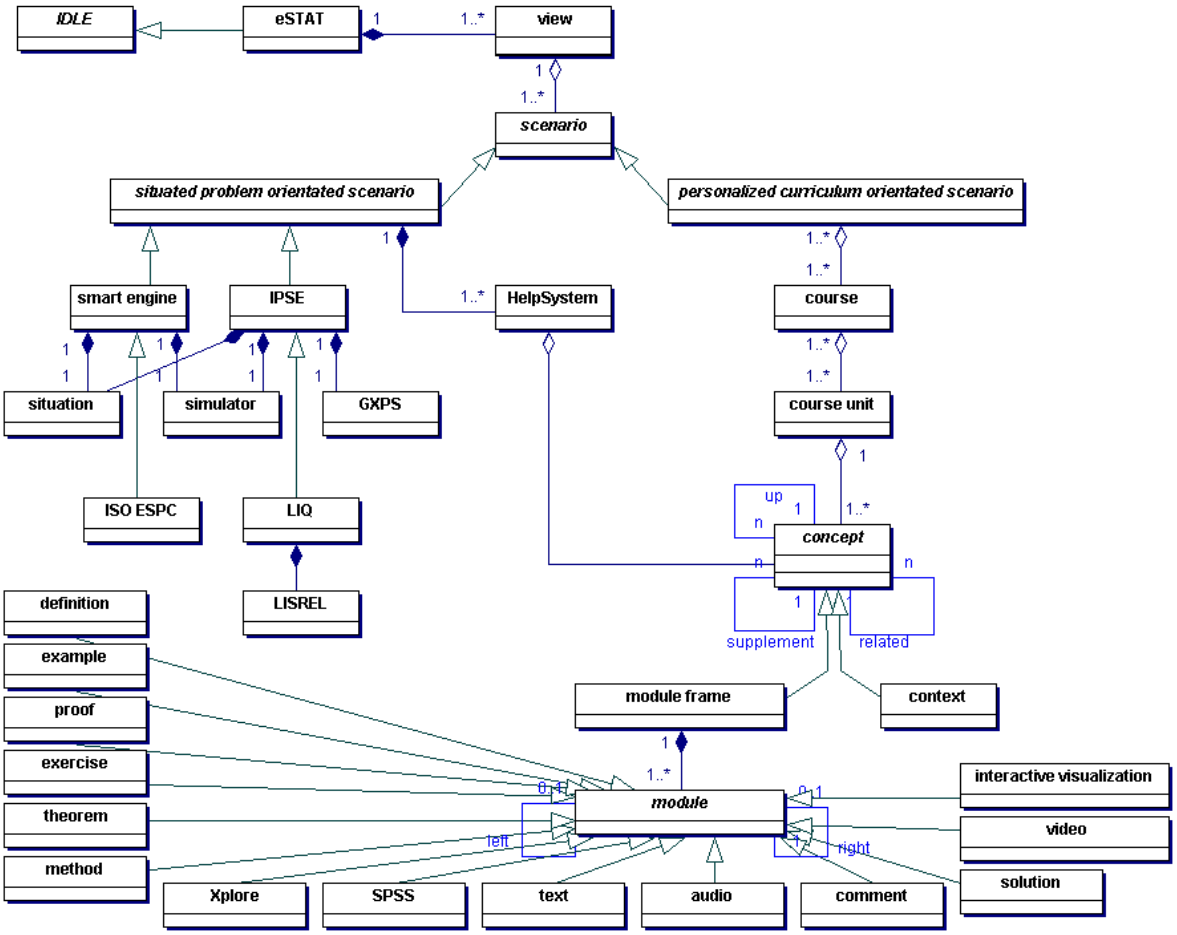

Figure 2: Class Diagram of Top Level of IDLE e-stat

They can be specialized by text blocks (“context”) or by module frames. Concepts can be linked by associations (“up”, “related” and “supplement”), which can be used to build a “quick and dirty” ontology by the statistical content providers. Module frames are again composite aggregates of modules, which are the smallest units of learning. Types of modules were provided to us by our domain partners at the local department of mathematics. Inside module frames, modules are linked to define the partial order „X depends on Y”. We hope that the meaning of the SIPOS-side of the class diagram is self evident when reading the next paragraph.
The class diagram was translated to a XML-DTD[31] to provide content providers with a structured editor for the nonabstract classes.

Specification of e-stat-IPSE “LI.Q.”

As a type of SIPOS we conceptualize and specify a further IPSE: „the LISREL-Quiz“. The question is „how high is your LI.Q?“. What is your competency in modeling stochastic parts of the world with LISREL (Linear Structural Equation) Models[32]. The scenario follows the IPSE specification on the meta level quite closely, though the required artifacts on the implementation level are completely different from those in former IPSEs. First a random sample of virtual persons with unknown attribute values is generated. These persons pass a chamber with digital gauges, which meaning is unknown. The stochastic interrelations of the gauges are compressed in a covariance matrix which is the specification of the task "Are the data congeneric?“ or „Do all gauges measure the same latent construct?" In another scenario we work with real data: module grades of bachelor students. To answer the question students have to develop the artifact E, which is in this case a LISREL-model. We can test the (stochastic) correctness T |= E with the LISREL-Program. If E does not implement the specification according to our IPSE-philosophy the student should present hypotheses concerning E to the IPSE. For this service we have to develop a special AI-component: eg. a cognitive model of an expert statistician. The full functionality of the IPSE

should be integrated in the IDLE. The necessary technical LISREL modeling knowledge can be pulled from the help system. This imports the same modules which are used in the PECOSs.

Standardization Efforts in eLearning

The eLearning standardization process is an active, continuously evolving process.
The IEEE's LTSC is the institution that is actually gathering recommendations and proposals from other learning standardization institutions and projects. Specifications that have been approved by the IEEE go through a more rigorous process to become ANSI or ISO standards.
We take a closer look at the Educational Modelling Language (EML) developed by the Open Universiteit Nederland (OUNL). The OUNL is actively involved in the development of standards and specifications with PROMETEUS, IMS and CEN/ISSS WS-LT.

Specification of eLearning with EML

During the work in our project we came across the EML learning technology specification[33]. A provisional definition is given[34]: ”EML is a semantically rich information model and binding describing the content and process within 'units of learning' from a 'pedagogical perspective'.".
EML is developed based on fundamental axioms about the way people learn. The pedagogical meta-model consists of four (conceptual) packages: (1) Theories of learning and instructions, (2) Learning Model, (3) Unit of Study Model and (4) Domain Model.
As mentioned above ISP-DL can be regarded as a specialization of the learning-model in EML. “Unit of study” is the most elaborate component in the design. It represents all content and processes in function of learning something. In order to be more precise a unit of study consists of meta data, roles, learning objectives, prerequisites, content, and methods. In addition to general information about the unit of study (e.g. title, author, creation-date) the meta data may contain a short description and the minimum/maximum time for completing that unit. The roles learner and staff can be modeled in EML. Learning objectives describe the intended goals. Prerequisite states the conditions under which a learner should start the unit of study. The construct content includes the environments and the activities. The latter are the indivisible tasks to be performed in the learning environments. Methods contain sequences of activity structures, plays (assignment of activities to roles), and conditions. They describe how the different learners progress through a sequence of activities in individual ways to achieve the learning objectives.
The study of EML shows that the PECOS-part of our IDLE may be specified by EML: eg. our class diagram can be mapped to EML. The question is whether we get the same insight for the SIPOS-part of the IDLE, especially for the IPSE component.

Mapping IPSEs into EML

We have to check whether it is possible to specify the knowledge based parts using EML. To answer this question we take a look at the IPSE artifacts. If EML can supply (or specify) the same artifacts, then we can assemble an EML-based AI-specification.
We differentiate between the IPSE artifacts and the learner generated artifacts. Table 2 shows the mapping of the IPSE specification to the artifacts. We want to demonstrate that despite the uniform specification on the meta-level the implementation details are quite different. Exemplarily the artifacts relating to the specification element T |-? E (row 4) are described. The proposal E is checked by an expert system whether it is deducible from the systems domain theory T by inference rules. In the IPSE ABSYNT the solution proposals are checked by gmr-rules, a kind of horn rules. Different to that in Petri-Help the solution proposals are investigated by model-checking. Therefore no inference rules are used in Petri-Help. In WULPUS the diagnostic component is implemented as a constraint net consisting of equations which could be implemented as inference rules. In the IPSE LI.Q. the LISREL program is applied to examine the correctness of the solution proposals. So, in the LISREL quiz no inference rules (or something similar) are utilized.
Table 3 displays the mapping of the IPSE specification to EML constructs. We want to discuss the mapping of the learners’ proposal E into EML “Activity-Sequences” and of the self-explanation process S' |- E' into EML “Learning Objectives” in more detail.
The Activity-Sequence has to describe the desired activities of a learner. Due to the large solution space in IPSEs this is an at last tedious or practical impossible task. For instance ABSYNT is able to recognize several million solutions to 40 tasks.
Another problem in mapping into EML shows the specification of our meta learning objective ‘self-explanation’. From a cognitive psychology point of view ‘self-explanation’ is a highly desirable process for a learner. Though our former IPSEs did not contain a special artifact to support ‘self-explanation’ (besides the capability to support hypotheses testing), IPSEs in progress (mile/ET and LI.Q.) will contain it. A look into the EML Reference Manual, shows that there is no simple way to describe a Learning Objective ‘Self-explanation’. The main problem is that EML is not expressible enough to specify AI or CS (Cognitive Science) motivated artifacts, learning objectives or results. For this reason EML is not useful for specifying IDLE systems.

IPSE specification	ABSYNT	Petri-Help	WULPUS	LI. Q.
S	--	--	--	Knowledge assessment?
S |= E, S |- E	--	--	--	Self-Explanation?
$S = S_{fix} \cup S_{mod}$	--	--	--	Knowledge assessment?
T |-? E	gmr	--	constraint net	--
T |=? E	--	model-checker	--	LISREL
$E = E_{fix} \cup E_{mod}$	editor	editor	input fields	Editor?
T |- E‘	gmr	Case-learner	constraint net	AI-heuristics?
$E' = E_{fix} \cup E'_{mod}$	gmr	Case-learner	constraint net	AI-heuristics?
S‘ |= E‘, S‘ |- E‘	--	--	--	Self-Explanation?
$S' = S \setminus S_{mod} \cup S'_{mod}$	--	--	--	Knowledge assessment?

Table 2: Mapping IPSE Specification to IPSE Artifacts

IPSE specification	EML construct	**but...**
S	Role=Learner (self-assessment)	Mental state
$S = S_{fix} \cup S_{mod}$	Role=Learner (self-reflection)	Mental process
E	Activity-Sequence	**too many** solutions
S |= E, S |- E	Role=Learner (self-explanation)	**no simple** specification possible
T; T |- E	Knowledge Object, Learning Objective	**no simple** specification possible
T |-? E, T |=? E	Tool Object	**no simple** specification possible
T $\neg$|- E; $E = E_{fix} \cup E_{mod}$	Activity Selection	**no simple** specification possible
T |- E‘	Tool Object; Conditions	**too complicated** for XML
$E' = E_{fix} \cup E'_{mod}$	Role=Learner (self-explanation)	Mental process
S‘ |= E‘, S‘ |- E‘	Learning Objective	**no simple specification possible**
$S' = S_{fix} \cup S'_{mod}$	?	Mental process

Table 3: Mapping IPSE specification to EML

Summary

To offer intelligent, learner adapted anytime services to learners it is necessary to use AI-components in DLEs. When the system is implemented by different developers and content providers it is necessary to specify all components and interfaces of the system to assure some consistency and quality. We used logic for meta-level specification of AI-components (IPSEs) and UML/XML for the rest of the IDLE. It turned out that one of the new eLearning specification languages (EML) is not powerful enough to specify the behaviour of the AI-components and the solution spaces in situated problem orientated scenarios. Worse, those languages restrict the knowledge and behaviour spaces of the learner to simple subspaces (eg. multiple choice tests). So the specification of eLearning systems with EML pushes the outline of the system more in the direction of PECOS-systems than in the direction of SIPOS-systems. Our future work will explore in what kind of direction we should progress: logic[35] or patterns[36] [37]?

References

[1] MICHALSKI, R.S., Toward a Unified Theory of Learning: Multistrategy Task-adaptive Learning, in B.G. Buchanan, D.C. Wilkins (eds), Readings in Knowledge Acquisition and Learning, 7-38, San Mateo, Calif.: Morgan Kaufmann Publishers, 1993

[2] 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

[3] 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

[4] MOEBUS, C., SCHROEDER, O. & THOLE, H.J., Interactive Support of Planning in a Functional, Visual Programming Language, in P. Brna, S. Ohlsen, H. Pain (eds), Proceedings AI-ED 93, World Conferences on Artificial Intelligence and Education, Edinburgh, 362-369, 1993

[5] MOEBUS, C., SCHROEDER, O. & THOLE, H.J., Diagnosis of Intentions and Interactive Support of Planning in a Functional, Visual Programming Language, in D.M. Towne, T. de Jong, H. Spada (eds), Simulation-Based Experimental Learning, Berlin: Springer (NATO ASI Series F: Computer and Systems Sciences, Vol. 122), 61-76, 1993

[6] ANDERSON, J.R., Knowledge Compilation: The General Learning Mechanism. In: R.S. Michalski et. al., Machine Learning II. Kaufman, 1986

[7] ANDERSON, J.R., A Theory of the Origins of Human Knowledge, Artificial Intelligence, 1989

[8] NEWELL, A., Unified Theories of Cognition, Cambridge, Mass.: Harvard University Press, 1990

[9] VAN LEHN, K., Toward a Theory of Impasse-Driven Learning, in H. Mandl et. al.(eds), Learning Issues for Intelligent Tutoring Systems, Berlin: Springer, 1988

[10] HECKHAUSEN, H., Motivation und Handeln, Heidelberg: Springer, 1989

[11] GOLLWITZER, P.M., Action Phases and Mind-Sets, in: E.T. Higgins & R.M. Sorrentino (eds), Handbook of Motivation and Cognition, Vol. 2, 1990

[12] BERCHT, M. & VICARI, R.M., Pedagogical Agents with Affective and Cognitive Dimensions. In: Congreso Iberamericano de Informatica Educativa. RIBIE, 5., 2000, Viña del Mar. Actas, Santiago, Universidade de Chile, 2000

[13] VICARI, R.M., ITS, Agents, BDI, and Affection: Trying to Make a Plan Come Together, Invited Paper, in: St.A. Cerri, G. Gouarderes & F. Paraguacu (eds), Intelligent Tutoring Systems, Proceedings of the 6th International Conference, ITS 2002, Biarritz, France and San Sebastian, Spain, June 2-7, 2002, Berlin: Springer, Lecture Notes in Computer Science, LNCS 2363, ISBN 3-540-43750-9, 2002

[14] MOEBUS, C., SCHROEDER, O., The Acquisition of Functional Planning- and Programming Knowledge: Diagnosis, Modelling, and User-Adapted Help, in G. Strube, K.F. Wender (eds), The Cognitive Psychology of Knowledge. The German Wissenspsychologie Project (Advances in Psychology Series), Amsterdam: Elsevier (North-Holland), 233-261, 1993

[15] RUSSEL, St.J., The Use of Knowledge in Analogy and Induction, Research Notes in Artificial Intelligence, London: Pitman, 1989

[16] BARROWS, H.S. & TAMBLYN, R.M. Problem-based learning: an approach to medical education, New York: Springer, 1980

[17] DUFFY, T.M., JONASSEN, D.H., Constructivism and the Technology of Instruction: A Conversation, Hillsdale: Erlbaum, 1992

[18] SCHULMEISTER, R., Virtuelle Universität – Virtuelles Lernen, München: Oldenbourg, 2001

[19] MÖBUS, C., SCHRÖDER, O., THOLE, H.-J., WULPUS - An Intelligent Problem Solving Environment Delivering Knowledge Based Help and Explanations in Business Management Simulation, in C. Frasson, G. Gauthier, A. Lesgold (eds), Intelligent Tutoring Systems, Proceedings of the Third International Conference ITS 96, Montreal, June 1996, Berlin: Springer (LNCS 1086), 1996

[20] THOLE, H.-J., MÖBUS, C., SCHRÖDER, O., Domain Knowledge Structure, Knowledge Representation and Hypotheses Testing, in: B. Boulay, R. Mizoguchi (eds.): Artificial Intelligence in Education Proceedings of AI-ED 97 World Conference on Artificial Intelligence in Education, Kobe, Japan, Amsterdam: IOS Press, 1997

[21] WILLMS, J., Konzeption einer intelligenten Problem-löseumgebung für die Patentanmeldung und –prüfung, Dissertation, Universität Oldenburg, Fachbereich Informatik, 2002, http://www.dissertation.de.

[22] YAKIMCHUK, V., GARBE, H., MÖBUS, C., THOLE, H.J., Eine intelligente Problemlöseumgebung für die Grundlagen der Elektrotechnik, 47. Internationales Wissenschaftliches Kolloquium, TU Illmenau, 23. - 26. 9. 2002

[23] DE RAEDT, L., Interactive Theory Revision, San Diego: Academic Press, 1992

[24] VAN LEHN, K, Mind Bugs: The Origins of Procedural Misconceptions, Cambridge, Mass.: MIT Press, 1990

[25] Förderantrag an das BMBF, Förderkennzeichen 08NM058A, 2000

[26] JANK, W & MEYER, H., Didaktische Modelle, Frankfurt a. M.: Cornelsen Scriptor, 1994

[27] BRUNS, B. & GAJEWSKI, P., Multimediales Lernen im Netz: Leitfaden für Entscheider und Planer, Berlin: Springer Verlag, 1999

[28] WARSITZ, B., Zur Strategischen Ausrichtung und Implementierung von E-Learning-Umgebungen, CML Services, 2002, http://www.cml-services.de/

[29] FORBUS, K.D., FELTOVICH, P.J., Smart Machines in Education, AAAI Press 2002

[30] BUNSE, Ch., von KNETHEN, A., Vorgehensmodelle Kompakt, Heidelberg: Spektrum 2002

[31] FENSEL, D., Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, Berlin: Springer, 2001

[32] CUDECK, R., du TOIT, S. & SÖRBOM, D. (eds), Structural Equation Modeling: Present and Future, A Festschrift in honor of Karl Jöreskog, Chicago, Ill.: Scientific Software International, Inc., ISBN: 0-89498-049-1, 2001

[33] KOPER, R., Educational Modelling Language: Adding Instructional Design to Existing Specifications, in: FHG & Uni Frankfurt (Hrsgb), Reader zum Workshop „Standardisierung im eLearning“, 10.-11.4.2002

[34] HUMMEL, H., Survey of Educational Modelling Languages, in Proceedings of the Learning Technologies Workshop CEN/ISSS, Koper, Rawlings, Lefrere, R. Artacho, Kovacs (eds), Report CEN/ISSS, 2002, in press

[35] REITER, R., Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems, Cambridge, Mass.: MIT Press, 2001

[36] DEVEDZIC, V., Applying Patterns to ITS Architectures, in: G. GAUTHIER, C. FRASSON, K. VAN LEHN (eds), Intelligent Tutoring Systems, 5th International Conference, ITS 2000, Montréal, Canada, June 2000, Proceedings, Berlin: Springer, Lecture Notes in Computer Science, LNCS 1839, ISBN 3-540-67655-4, 123–132, 2000

[37] DEVEDZIC, V., & HARRER. A., Architectural Patterns in Pedagogical Agents, in: St.A. Cerri, G. Gouarderes & F. Paraguacu (eds), Intelligent Tutoring Systems, Proceedings of the 6th International Conference, ITS 2002, Biarritz, France and San Sebastian, Spain, June 2-7, 2002, Berlin: Springer, Lecture Notes in Computer Science, LNCS 2363, ISBN 3-540-43750-9, 81-90, 2002