ACE - Adaptive Courseware Environment

Marcus Specht and Reinhard Oppermann
GMD - German National Research Center for Information Technology
Institute for Applied Information Technology (FIT-MMK)
D-53754 Sankt Augustin
email: {Marcus.Specht, Reinhard.Oppermann}@gmd.de

Abstract: The Adaptive Courseware Environment (ACE) is a WWW-based tutoring framework which combines methods of knowledge representation, instructional planning, and adaptive media generation to deliver individualized courseware via the WWW. ACE is based on a domain model of the subject matter, a pedagogical model on how to teach a curriculum, and learner modeling on different levels, e.g., preferences, interests, and knowledge. Based on these three components individualized web-content (html, java, pictures) is generated and presented to the learners. Taking into account the interest and the knowledge of a learner ACE can adapt different aspects of the instructional process, e.g., adapting the curriculum by selection of content, adapting the presentation of contents by choosing appropriate media and combining them, adapting the teaching strategies for specific contents, annotating hyperlinks, and by recommending appropriate hyperlinks and contents. Currently three systems have been realized on the basis of ACE and adaptive components of each system have been evaluated in empirical studies. Additionally experimental studies of the applied adaptive methods have shown improvements of efficiency and effectiveness of learning compared to classical static hypermedia.

1. Introduction

WWW-based educational systems are becoming increasingly popular and are used by heterogeneous user groups. Users of educational WWW-based systems differ in their goals, professional background, interests, and knowledge of the subject matter. To optimize the efficiency and effectiveness of the learning process educational systems should adapt to these user characteristics.

Adaptive methods and techniques in learning machines have been introduced and evaluated since the 1950's in the area of adaptive instruction and the psychology of learning. Adaptive instructional methods adapted the content of the instruction [1], the sequencing of learning units [2, 3], the difficulty of units [4], and other instructional parameters to the students' knowledge. For a overview see [5]. These methods have been empirically evaluated and shown to increase learning speed and to help students gain a better understanding through individualized instruction. Furthermore, the integration of AI methods [6-8] can enhance adaptive methods in computer based learning.

In the following we introduce the ACE framework which utilizes hypermedia technologies and AI methods to deliver individualized hypermedia instruction. In the first part of the paper we will give an overview of related works that led us to the ACE framework presented in this paper. In the second part the architecture is introduced and the relations between the three main components are described. Third we describe the processes of authoring, generating, and delivering individualized courseware with ACE. Finally we will give an overview of implemented adaptive methods and the results of empirical evaluations done within ACE.

2. Related Works

The current version of ACE is motivated by several works in the field of adaptive hypermedia. The ELM-ART II [9] system transferred an intelligent on-site learning environment to an intelligent integrated textbook on the WWW. The system basically is a starting point for the knowledge representation approach of ACE. ELM-ART II uses a concept based knowledge representation and prerequisite relations between the units to guide the student through the learning space. A similar approach can be found in INTERBOOK [10]. However ACE tries to enrich a concept based knowledge representation with the integration of different learning materials and their pedagogical role in the teaching process. Comparable approaches for adaptive selection of teaching activities in intelligent tutoring systems can be found in GUIDON2 [11]. Furthermore the integration of different modalities and their combination in adaptive hypermedia presentations is motivated by works of Feiner &McKeown [12] or Goodman [13].

The learner model of ACE combines a probabilistic overlay modeling [14] and episodic approach [15] of learner modeling. The probabilistic overlay model can be used for several adaptive methods like adaptive sequencing, mastery learning, adaptive testing, and adaptive annotation. By storing individual interaction episodes and cases with each unit, episodic information is added to the probabilistic learner model. These interaction episodes are used to generate hypotheses about the learner's knowledge and interests and negotiate them with the learner.

Within ACE several adaptive methods have been implemented and evaluated. These adaptive methods can be categorized and described according to Brusilovsky's overview [16]. The main focus in the evaluation of adaptive methods within ACE is on methods of adaptive navigation support, which is one of the most popular methods in current adaptive hypermedia systems like ELM-ART II [9], INTERBOOK [10], or AHA! [17]. Motivated by empirical results the adaptation of adaptive methods has been integrated in ACE, i.e., for example the adaptive annotation of hyperlinks can be adapted to several changing characteristics of learners (knowledge, interests, history). On the one hand this allows authors of adaptive hypermedia to specify adaptations on a high level (see Figure 4), on the other hand this reflects the empirical results that the successful application of adaptive methods in hypermedia is often confounded with learner characteristics like the previous knowledge of learners. The experiences with different curricula implemented in the ACE framework [18, 19] resulted in the integration of adaptive components with varying restrictions on learner's freedom and adaptation to different learner features. In the statistics curriculum of AST [19] and adaptation to a clear sequence of learning units, the prerequisite relations and the learners knowledge about the units appeared to be sufficient. In a less structured domain like the curriculum on narcotics in ADI [18] adaptation of recommendations based on interests, the learner's previous interaction episodes, and the learner's media preferences seemed to be more adequate. In the ACE framework the author of a new curriculum can integrate these different adaptive components on a high level and administrate them to support the learner in an adequate way.

3. A knowledge based architecture for individualized learning

Psychological models of teaching often mention three sources of knowledge used by competent teachers [20]. First, teachers are experts in the subject matter (e.g., they know about the concepts of the domain and their interrelation, are able to critique solutions of problems, answer questions, give examples, and far more). Second, teachers know how to teach something (e.g., they use strategies to teach a concept, they know when to use a certain teaching material or presentation method). Third, teachers build a model of the students' knowledge. This allows teachers to adapt their teaching methods and goals to different students or groups of learners.

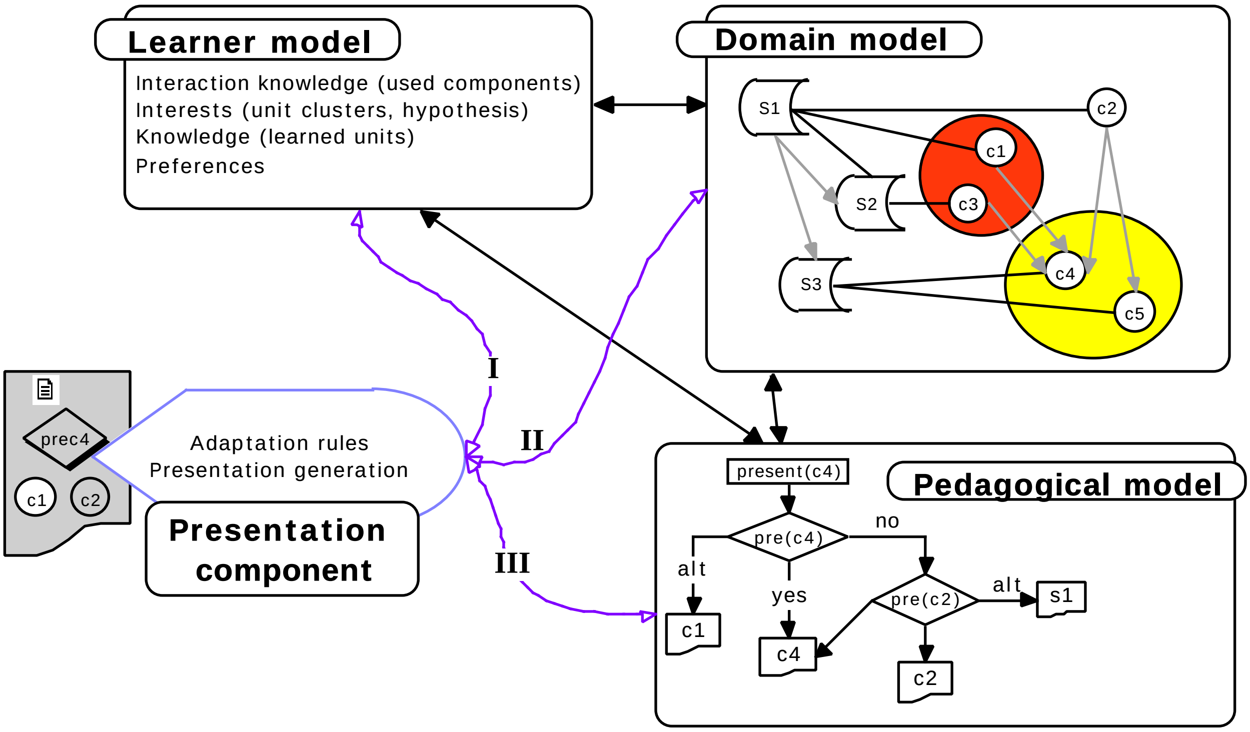


Figure 1: The ACE architecture and its main components.

Most classical Intelligent Tutoring Systems (ITS) [21] include aspects of knowledge representation, tutorial components and learner modeling in the upper sense to achieve the goal of individualized instruction. Accordingly, the ACE architecture shown in Figure 1 consists of three main models:

The domain model describes the concepts of the domain and their interrelations and dependencies, the **pedagogical model** contains pedagogical strategies and diagnostic knowledge and the **learner model** stores the preferred settings of a learner, the domain concepts a learner worked on, and the interface components used by the learner. We will discuss each of these models in turn.

3.1. The domain model

The domain model contains the knowledge about the domain and the curriculum structure. Basically, the model is built on a conceptual network of learning units. Learning units can either be sections or concepts. While sections can contain other units, concepts can not. A small sample of a domain model with three sections and five concepts can be seen in figure 1. The lines in the graph represent container relations for a default sequence specified by the author of the curriculum, the arrows stand for prerequisite relations. In the example concept number 2 (c2) is a prerequisite for the concepts number 4 and 5 (c4, c5). The distinction between concepts and sections reflects the different roles of units in the instructional process. Sections provide information about grouping and ordering of domain concepts, they give an introduction or sum up the information given by a group of units. Concepts hold one unit of knowledge and explain different aspects of it with different types of learning material. The different kinds of learning material are text (introduction, several levels of text, summary), examples, demonstrations, interactive playgrounds, and tests. Each of these learning materials as such has a different pedagogical role in the teaching process. An overview is given in Table 1.

Table 1: Learning materials and their pedagogical functionality

Learning material	Pedagogical functionality
Introduction	The introduction should give an overview of a learning unit and allow the learner to put the unit into relation with his current knowledge.
Text	Texts give detailed information on several levels of complexity and about different aspects of the learning concept.
Example	Examples give concrete instantiations or contextual descriptions.
Animation	Animations allow the student to watch interdependencies or relations between different concepts
Simulation	Playgrounds allow the students to play around and acquire knowledge and skills necessary to fully understand a learning unit. In difference to animations, learners can actively influence simulations.
Test	Tests allow checking of the level of knowledge the student currently has. A student can also use tests for self-assessment.
Summary	Summaries should summarize the facts and skills taught in a concept.

In addition to the units and the associated learning materials, several other relations can be specified between units in the domain knowledge network.

- Implicitly, the author of a curriculum specifies the **curriculum sequence** relation. This relation is used to give the student a default path through the curriculum. The default sequence is specified by a serialization of the sections and their subsections and concepts.
- **Prerequisite** units describe units the learner should be familiar with before working on a unit. Each prerequisite unit is weighted according to its importance as a prerequisite. This relation is comparable to [22] and is used for the adaptation of the unit sequence. The prerequisite relations between units specify a partial order of units and can be combined with the default curriculum sequence to individualize the students learning sequence depending on their current knowledge.
- **Interest** fields identify clusters in the conceptual domain network. For instance, in figure 1 the concepts c2 and c3 are in one interest cluster and the concepts c4 and c5 are in another

interest cluster. Mainly with each concept one or more fields of interest can be specified and the unit can be weighted with a relevance value between 0 and 1. These relations can be used for individualized recommendations based on the student's interest model like described in section 5.3.

3.2. The pedagogical model

The pedagogical model represents the teacher's knowledge of how to teach units. It consists of two main parts: teaching strategies and diagnostic knowledge.

3.2.1. Teaching strategies

Teachers follow different strategies to teach different types of concepts. Teaching strategies in this sense are comparable to the teaching rules in Vassileva [23]. For example, when teaching statistics, introducing the arithmetic mean can easily be done with a text, a formula, some tasks, or an interactive playground. In contrast, it is very hard to "explain" a more abstract concept with a text or a formula. For introducing a more abstract concept, e.g., scale levels, it would be much easier to start with concrete examples and then go on with a more abstract text definition or a formula. Such different strategies in the sense of a certain sequence of learning materials are defined in the teaching strategies. The system has a default strategy for each concept and authors can specify a preferred strategy for each concept if needed. Rules are defined with each strategy, which allow the system to adaptively modify a teaching strategy depending on critical learner features. For example, when a student requests information about the concept "categorical data" the system retrieves the teaching sequence for this concept specified by the author, e.g., first give an example, second the text for the concept, and tests at the end. Modifications of this presentation sequence can be triggered by the contents of the learner model. For example if the student has already seen some of these learning materials they will be skipped, or if the student has already worked on tests about this concept successfully the system will modify the presentation and just present some advanced examples and the summary about this concept. Furthermore, if the system detects prerequisites for the requested concept that the learner did not work on, the system integrates a diagnostic module for the prerequisite units in the presentation of the current unit. In Table 2 some examples for the rules implemented in the system and the resulting presentation can be seen.

Table 2: Example modifications of the default teaching strategies depending on learner characteristics

Learner characteristic	Teaching sequence, Presentation
The learner did not work on the current unit and has specified no media preferences.	Default material sequences, no modifications.
The learner did not work on the current unit and has specified media preferences.	Default material sequence with the preferred material at the beginning.
The learner is not familiar with prerequisites of the current unit.	Warning, introduction and hyperlink to prerequisite testing.
The learner has done some tests on the current unit with less success and already seen some learning material.	Mixed sequence of missing learning material, easy tests, and repetition of texts.
The current concept is already learned by the student. The system has tested his knowledge.	Summary and hyperlinks to units that have the current unit as prerequisite.

3.2.2. Diagnostic knowledge

One problem with learning environments on the WWW is the reduced bandwidth of information. From the fact that a student loaded a text page no valid inferences can be drawn about the actual knowledge of the student. Integrating interactive diagnostic modules in a WWW-based learning environment is one more valid way of getting information about the learner’s knowledge. The diagnostic components store the knowledge about several types of tests and how they have to be generated and evaluated similar to Nkambou and Gauthier’s [24] pedagogical resources. Each test in the test base can be connected to multiple units and is weighted on difficulty and relevance for these units. Depending on the difficulty and the relevance of a test, a student’s answer has different impacts on the learner model. In addition to knowledge tests the ACE framework integrates diagnostic modules for background interests and previous experience with computers.

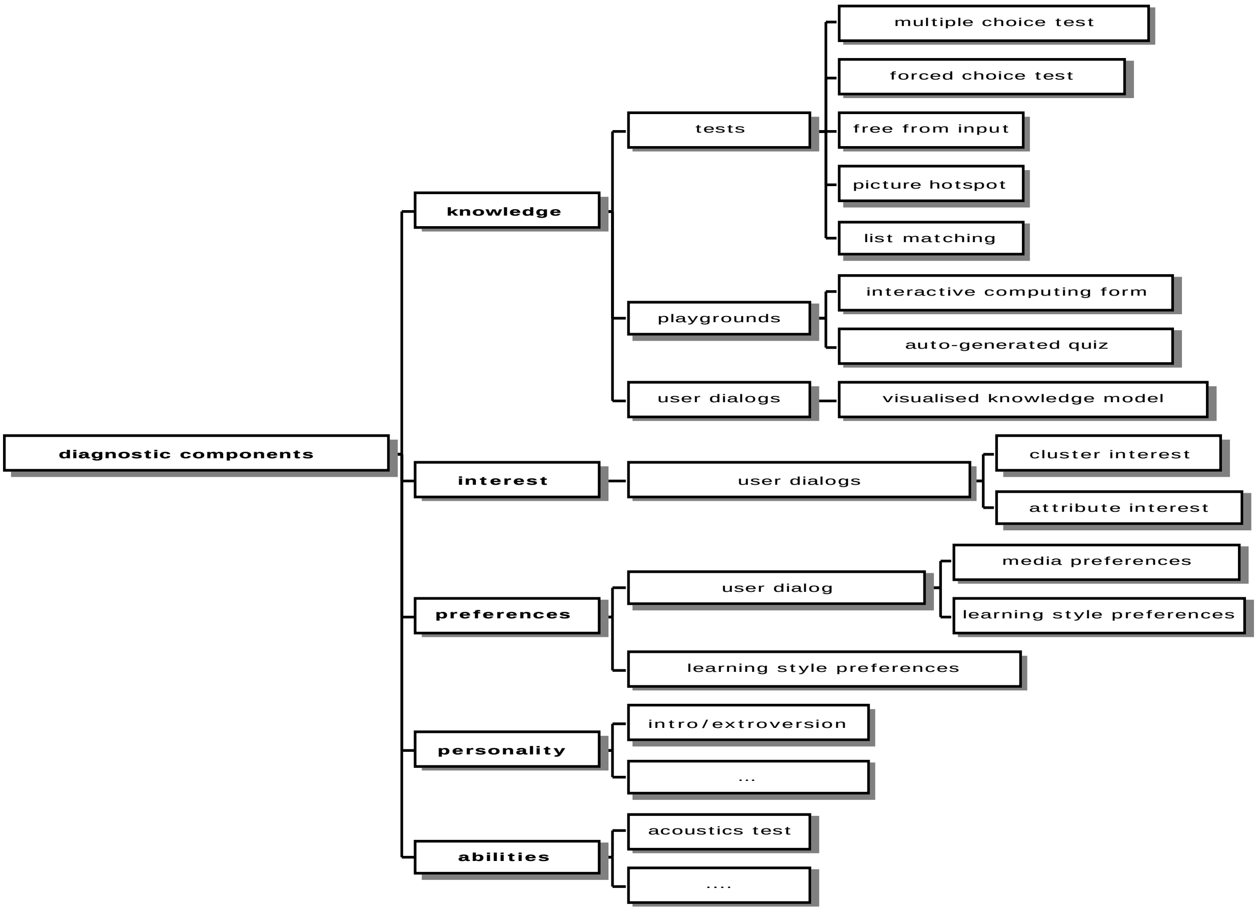


Figure 2: Diagnostic modules for different learner characteristics in ACE

Modules for cognitive abilities, and personality factors are foreseen but are not used in the current systems. Currently most of these diagnostic modules are realized in a very simple way (questionnaires, user dialogs). In a next step the diagnostic modules should be expanded to adequately reflect the diagnostic standards and validity criteria of psychological diagnostics. Figure 2 gives an overview of diagnostic modules in ACE.

3.3. The learner model

The learner model consists of three main parts: the learner settings, the knowledge model, and the interest model.

The learner settings store information about the learner preferences for language, media, interface settings, personal annotations, and current courseware booked by the student. The learner settings are currently updated through a direct user dialog. Learners' can request and modify their profile anytime.

The units a learner worked on are stored in the learner knowledge model as learned units. Learned units have confidence values depending on the experiences of a learner with a unit. The representation distinguishes between the tested confidence, the inferred confidence, and the requested confidence for a unit.

- The tested confidence is only updated by test results of a student. When a student passes a test the difficulty value of the test and the relevance for the current unit (an example can be seen in Figure 3) are taken into account for updating the confidence values. Furthermore the update function takes into account the performance in already passed tests and the number of tests available.
- The requested confidence is updated if a learner requests information of any type. The update function just counts the number of available learning material contributing to the concept and divides it by the number of available learning material.
- The inferred confidence is updated by inferences of the system, e.g., if a student solves all tests for a section the system infers that the student is also confident with the sub-sections and the sub-concepts of this section.

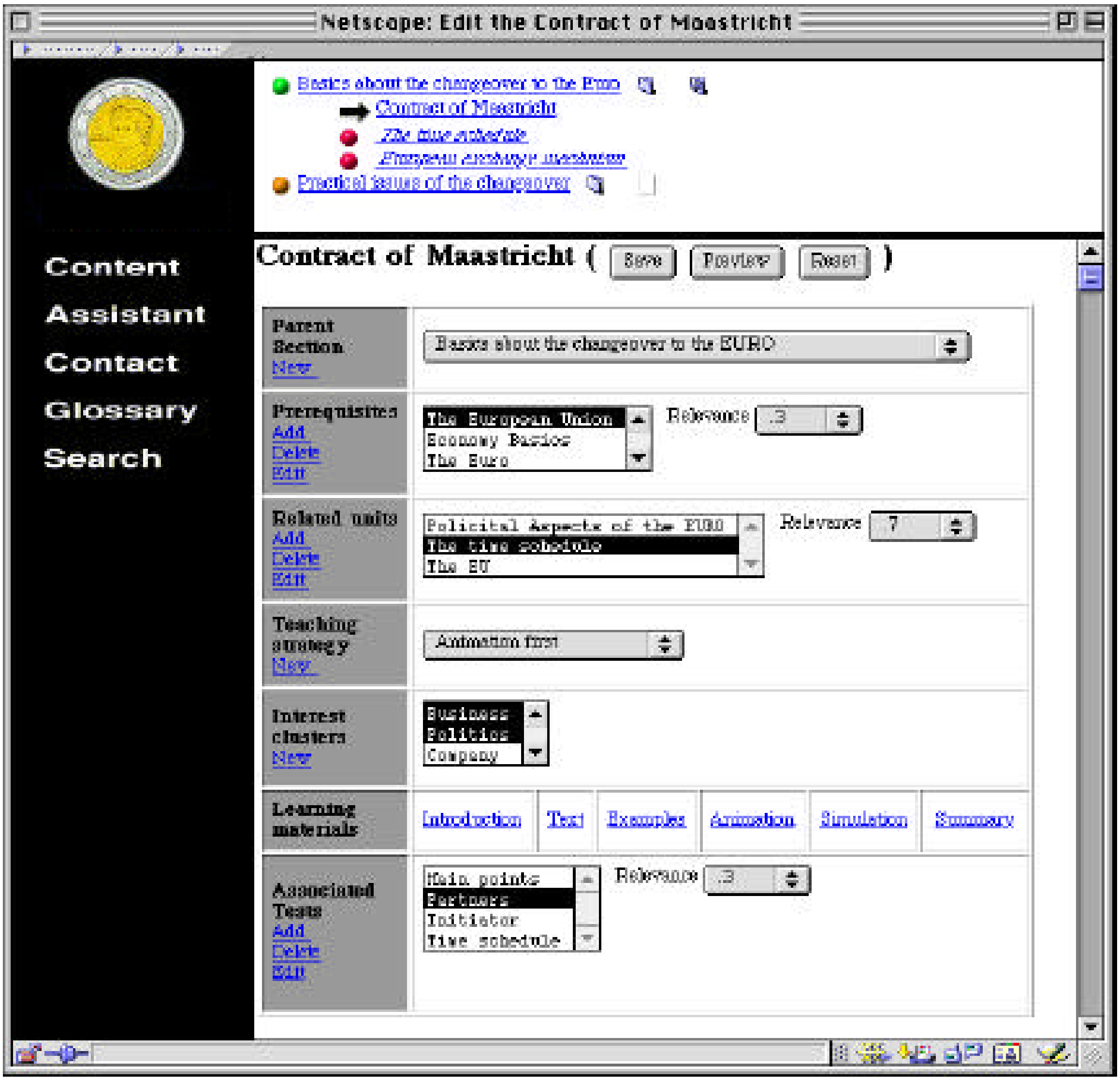
The distinction between different confidence values is used for different styles of interaction with the student during the learning process. Tested confidence values can be used for more restrictive teaching styles, i.e., the learners need to prove certain knowledge in tests to go further in the curriculum. The requested confidence values and inferred confidence values are used to generate assumptions about the learners knowledge and performance and negotiate them with the learners, i.e., displaying hypotheses about inferences of the system or visualizing a learner model and asking the learners to validate them. An example can be seen in Table 2.

The interest model contains the interest clusters a learner is interested in and dynamically builds hypotheses about the learner's interests. Hypotheses are build by rules that connect characteristics of episodes with certain assumptions about the learner's interests. For example if a learner requests a certain percentage of units from the mathematical cluster the system builds the hypothesis that the student is interested in the mathematical cluster and will present recommendations for related units in the curriculum.

All interactions of a learner with the objects of the curriculum are represented in interaction episodes. These interaction episodes are the input to the learner model. They contain a time stamp, the identifier of the unit involved, the material presented, material specific extensions, and the modality a material was presented in to the learner (text, audio, video). Whenever the learner requests an object or gets a presentation an interaction episode is sent to the learner model and the corresponding part of the learner model is updated.

4. Authoring and delivering adaptive courseware

Authoring courseware for the WWW confronts an instructor with a lot of problems. Among them: translating classical courseware into hypermedia courseware, selecting and preparing appropriate media, and delivering courseware within a server environment.



● Basics about the changeover to the Euro
→ Contract of Maastricht
● *The time schedule*
● *European exchange mechanism*
● Practical issues of the changeover

Contract of Maastricht (Save Preview Reset)

Parent Section New	Basics about the changeover to the EURO
Prerequisites Add Delete Edit	The European Union (selected) Economy Basics The Euro Relevance .3
Related units Add Delete Edit	Policital Aspects of the EURO The time schedule (selected) The EU Relevance 7
Teaching strategy New	Animation first
Interest clusters New	Business (selected) Politics (selected) Company
Learning materials	Introduction · Text · Examples · Animation · Simulation · Summary
Associated Tests Add Delete Edit	Main points Partners (selected) Initiator Time schedule Relevance .3

Figure 3: The editor for the concept “Contract of Maastricht”. Authors specify the prerequisites and related units with weights, select a teaching strategy, interest clusters, can edit the learning materials in an extra window and weight the connected tests. The difficulty of the tests is stored with the tests themselves.

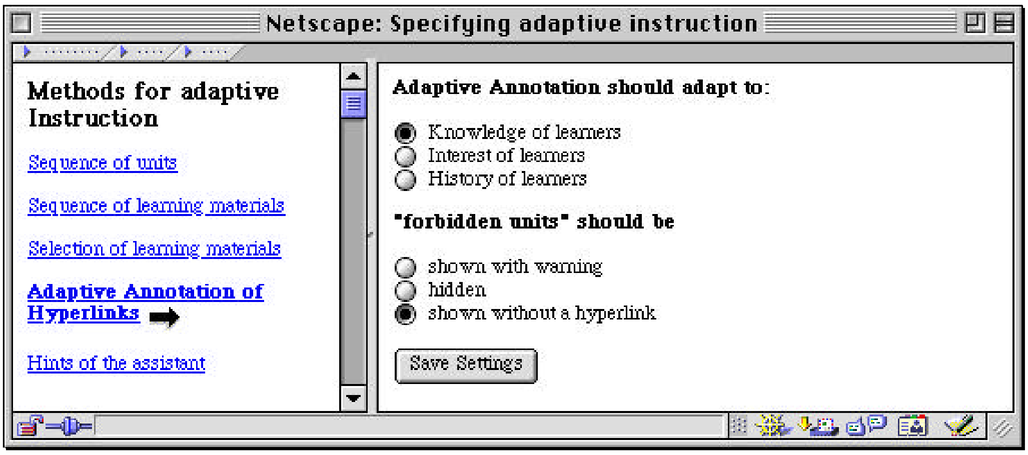
The ACE Editor is an easy way to modify and create adaptive courseware for the WWW. With the ACE editor authors can describe courseware based on the domain model of ACE and reuse and organize sections and concepts in new curricula. Additionally, authors can specify rules for the adaptive instruction of curricula. The ACE Editor expands the possibilities of currently available WWW authoring tools in the sense of adaptivity and instructional parameters that are specifiable by the instructors. Furthermore it shows a way of how to release instructors from the burden of specifying low level parameters about instruction with each unit of a curriculum. The instructors only specify general rules for the adaptive instruction and the ACE system applies these rules to the specialized and individual learning situations of students.

4.1. Creating and modifying courseware content

Basically, the ACE editor uses the same interface as the ACE learning environment. On the one hand this tries to give teachers an idea how the course will look for the students, on the other hand students that already worked with ACE can easily create new courseware. An example of editing a concept can be seen in Figure 3. The authors do not need to know any programming language to define a curriculum. Basically, the authoring process follows four steps:

1. *Define a curriculum structure with section and concepts*: With the interface shown in figure 3 authors can add and remove sub-sections and sub-concepts by simply clicking an icon in the content overview.
2. *Enriching the learning units with learning materials*. After selecting a learning unit the different learning material (introduction, texts, playgrounds, examples, tests, summary) can be specified by input forms. Additionally graphics, animations, videos and predefined presentations can be uploaded to the server where the curriculum and the material are stored.
3. *Specifying relations between the learning units*: authors are free to specify prerequisite relations, similarity relations, and interest clusters for their curriculum. The more relations are specified by the authors, the more adaptive support can be given to the learners.
4. *Specifying the instructional parameters*: From our point of view the main problem of currently available tools for adaptive instruction is the level of parameters and type of rules instructors have to specify for adapting the instruction. In most current systems instructors have to specify consequences depending on the learners actions with each unit or test in a curriculum. In a sense this is comparable to programmed instruction, where each next step of the adaptive instruction has to be planned by the instructor. With the ACE editor we try to give the instructors the power to specify adaptive instructional methods while not having to care about single learner interactions with certain learning units.

Figure 4 gives an example of how to specify the instructional methods for a whole curriculum.



Netscape: Specifying adaptive instruction

Methods for adaptive Instruction

Sequence of units

Sequence of learning materials

Selection of learning materials

Adaptive Annotation of Hyperlinks ➡

Hints of the assistant

Adaptive Annotation should adapt to:

● Knowledge of learners
○ Interest of learners
○ History of learners

"forbidden units" should be

○ shown with warning
○ hidden
● shown without a hyperlink

Figure 4: Editing a high-level rule for the annotation of learning units. The instructor specifies, that the criteria for annotation is the knowledge of the learner and that forbidden units should not be linked.

The authors just need to specify the learner feature the adaptive method should adapt to. As is shown in figure 4 the author specifies that the adaptive annotation of hyperlinks should adapt to the learner’s knowledge and forbidden units should be shown to the learner but should not be linked. From our point of view one critical question for the successful integration of adaptivity in authoring tools for learning environments is the level of adaptive parameters that instructors and courseware authors need to specify.

4.2. Delivering individualized courseware

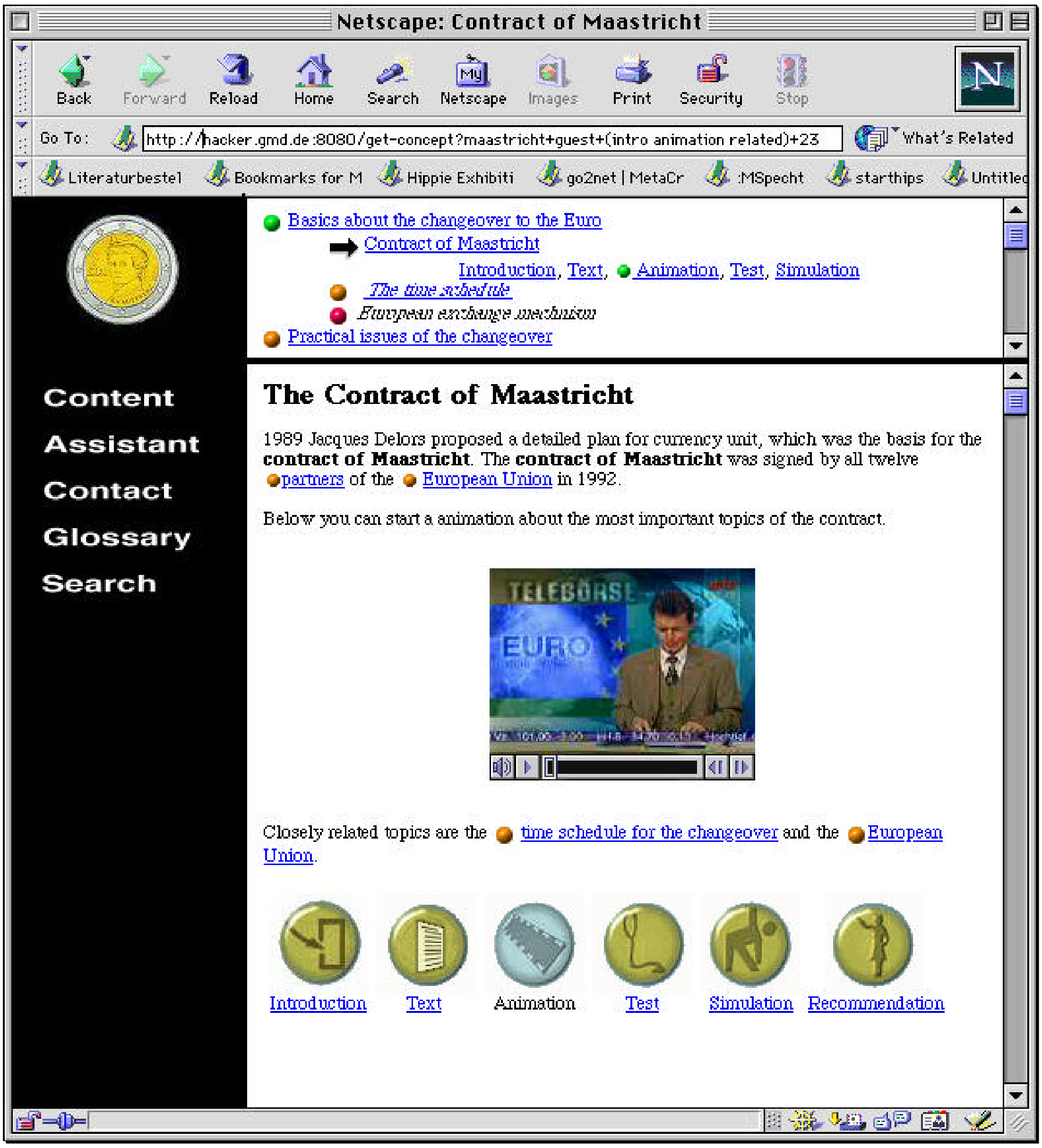
Currently several courses are realized on the basis of ACE. A course on introductory statistics and a curriculum on narcotics have been evaluated at the University of Trier [25]. A curriculum on business communication and a prototype about the changeover to the EURO are currently under development. With the different domains different types of diagnostic modules, teaching strategies and adaptive methods have been and will be evaluated.

Basically when students log into a course based on ACE they get an individualized homepage, where they can book courses, contact courseware authors and other learners, maintain and update their profile, and have a collection of resources linked to that page. From their personal ACE page they start working on the booked courseware.

When a student starts to work on a new courseware the system starts up with the diagnostic modules for learner preferences, interests and domain knowledge for the current curriculum. Students are asked for their preferences about learning material and general interests in an introductory questionnaire. Learners can specify preferred learning strategies like learning by example, reading texts, or learning by doing. In addition to that learners can estimate their expertise (novice, experienced, expert) in the domain. To validate the learner’s self-assessment ACE presents a short knowledge test dynamically generated from the domain model of the current curriculum, i.e., by using the default goal of the curriculum the system gathers all first level prerequisite concepts and their associated tests. The results from these knowledge tests and the other diagnostic modules are used to initialize the learner model.

Based on the learner model, the domain model, and the pedagogical model the presentation component of ACE selects appropriate learning units and generates individual hypermedia documents like shown in Figure 1. When a learner requests information about a learning unit, the ACE system checks if the learner has already mastered the requested unit or has seen some material about this unit (I). In a second step the system looks up relations and available material for the requested unit (II). In a third step the presentation component retrieves a plan for presenting the unit depending on the available learning material for this unit, the knowledge of the learner, and the teaching rules and pedagogical specifications of the course author (III).

The retrieved plan describes the presentation of the requested unit, which is translated into HTML by the presentation component. A plan and the corresponding multimedia presentation can be seen in Figure 5. Throughout the whole course the learner’s knowledge is tested so that the system can adapt to the dynamically changing knowledge, interests, and preferences of the learner.



http://hacker.gmd.de:8080/get-concept?maastricht+guest+(intro animation related)+23

● Basics about the changeover to the Euro
➡ Contract of Maastricht
Introduction, Text, ● Animation, Test, Simulation
● *The time schedule*
● *European exchange mechanism*
● Practical issues of the changeover

The Contract of Maastricht

1989 Jacques Delors proposed a detailed plan for currency unit, which was the basis for the **contract of Maastricht**. The **contract of Maastricht** was signed by all twelve ●partners of the ● European Union in 1992.

Below you can start a animation about the most important topics of the contract.

Closely related topics are the ● time schedule for the changeover and the ● European Union.

Figure 5 shows the presentation for the concept “Contract of Maastricht” depending on the settings in Figure 3 and Figure 4. The integrated learning materials are the introduction, an animation/video, and related units. The plan can be seen from the URL in the location bar of the browser. The arguments of get-concept are the unit identifier, the learner identifier and the learning materials’ identifiers.

5. Adaptive Methods in ACE

In ACE several adaptable and adaptive methods have been implemented and evaluated. The main groups of the implemented adaptive methods are in the field of adaptive navigation support, adaptive sequencing, adaptive recommendation, and pedagogical agent support. Furthermore experimental studies have been carried out on critical design issues of ACE. Central questions have been the user interaction with adaptive components and the impact of adaptive methods on learning time and knowledge gain [25].

5.1. Adaptive navigation support

Adaptive navigation support [16] is one of the most popular applications of adaptive presentation to the hypertext metaphor. ACE currently implements two methods of adaptive navigation support: Adaptive annotation and incremental linking of hyperlinks. These two methods of adaptive navigation support differ in their impact on learners' freedom to navigate in a hyperspace. Furthermore, a combination of these adaptive methods can restrict the users of a system in several ways.

5.1.1. Adaptive annotation

Adaptive annotation of hyperlinks supplies the user with additional information about the content behind a hyperlink. The annotation is adapted to the individual user by taking the learner's knowledge and relations of the units to be learned into account. In ACE the information about the learner's knowledge state and a pedagogical model of the domain units are used to adapt the presentation of colored balls next to each hyperlink. Concepts which have been visited by a learner are annotated with a hook. Concepts which the learner is not ready to learn (because of lacking prerequisites) are presented with a red ball, recommended concepts are annotated with a green ball, and hyperlinks to units that are not recommended but have no lacking prerequisites are shown with an orange ball.

Implementations of adaptive annotation can be found in different variants in [9, 17, 26, 27]. Furthermore also several experimental studies [28, 29] have investigated on the effects of these variations. The current form of adaptive annotation implemented in ACE [19] showed to increase learning speed and that students gain a better understanding of learned contents when combined with incremental linking (see 5.1.2). Furthermore a central finding was that more open forms of adaptive annotation are preferred by learners with higher prior knowledge, while students with low prior knowledge seem to be more active and gain a better understanding in more restrictive forms of adaptive annotation [30].

5.1.2. Incremental linking

In the area of adaptive interfaces the problem of overload in complex interfaces is often solved with the adaptive method of incremental interface. Elements of the interface and tasks manipulating these elements are incrementally introduced to a learner according to their difficulty and complexity and already mastered objects. The multitude of links in static hypertexts could cause a similar problem like complex interfaces. The floundering effect in hypermedia in that sense could be caused by an overload of too many alternative ways to proceed. Hyperlinks in learning materials can be integrated stepwise to reduce this overload. Texts in ACE can contain keywords to related concepts that are not learned by a student yet. If a text is presented to a student first, it will contain no hyperlinks to the related but not yet learned concepts. After working with the system, ACE will present all text keywords with hyperlinks to already mastered concepts and ready-to-learn concepts. Building an individual index in the learner model allows the system to present exactly these links in a text a learner is ready to visit, and units that are too complex are presented as plain text.

The content frame of ACE (see Figure 5) is an example of combining adaptive annotation and incremental linking. All units in the content overview are annotated with colored balls, which reflect the state of the unit in the sense of adaptive navigation support. Learned units are annotated blue, units the students lack prerequisites are colored red, and recommended units are shown green. Additionally, units that are annotated red are “forbidden” for the learners. By applying incremental linking based on the learner knowledge model the system disables hyperlinks for “not ready to be learned units”. Furthermore, ACE annotates the next best unit with an arrow and the learning materials of this unit are shown in detail. The links to the different learning materials are sorted according to the active learning strategy. As described in section 4.2 ACE provides meta-level facilities for authors and teachers to choose the form of adaptive annotation.

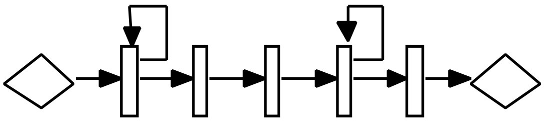
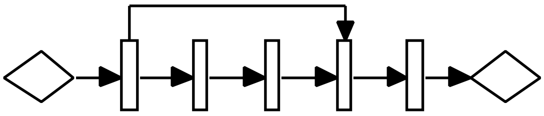
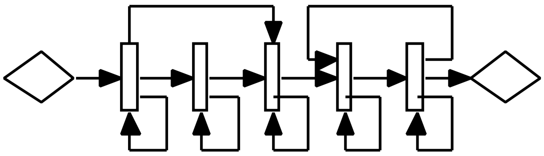
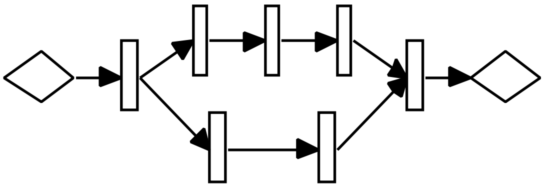
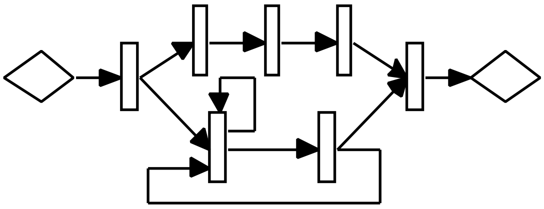
Incremental linking is a very restricting form of adaptive guidance for learners. Experimental studies within ACE showed that the successful application of incremental linking is strongly dependent on students' learning style and their prior knowledge [31]. Following these studies a major criteria for the adequate implementation of adaptive navigation support in learning environments appears to be the right balance between navigational support through annotation and guidance through incremental linking.

5.2. Adaptive sequencing

Salomon [32] differentiates three heuristic models for adaptive methods in learning environments: overcoming knowledge deficits, compensating more general deficits, adapting to preferences. Through a combination of different categories of sequencing algorithms specified by Frank [2] adaptive sequencing can be used in all three models of Salomon.

Table 2 gives an overview of adaptive sequencing algorithms used in ACE for different teaching goals. Adaptive sequencing can help to overcome knowledge deficits by adapting the complexity of learning material, compensate more general deficits through alternative learning sequences and adapt to individual preferences by selecting units or media. In ACE the sequence of learning steps is adapted on two main levels, e.g., the system gives proposals for sequencing of whole learning units and the system adapts the sequence of media available for a unit. While the sequence adaptation on the unit level mainly aims at overcoming of knowledge deficits, the sequencing of learning materials follows the preference model of Salomon [32].

Table 3: Sequencing algorithms used in the ACE framework

Sequencing algorithm	Description	Teaching goal
	Iteration: repeating units with different learning materials. An example for this can be seen in Table 2.	Overcoming knowledge deficits for one unit.
	Skipping units. If a learner successfully absolves tests on a section level, the subunits will be skipped.	More efficient learning, adapting level of complexity to students knowledge for better motivation
	Looping, skipping and iteration. This is a combination of the upper two. Furthermore the prerequisite warning is an example of this method.	Adapting to student preferences on unit and learning material level, fulfilling all prerequisites for learning units
	Alternative ways. An example of this is realized in the recommendations of the adaptive agent (see section 5.3)	Compensation of complex deficits, alternative learning strategies for units
	Alternative ways with looping, skipping and iteration. This is the most complex learning path adaptation, which can be found in combined adaptations of unit sequence and learning material sequence.	Overcoming knowledge deficits, compensation of complex deficits, adapting to media preferences, and fulfilling all prerequisites for learning units

5.2.1. Adaptation of unit sequence

The adaptive sequencing component tries to keep the student on an optimal path based on the student’s current knowledge. Furthermore, it adapts the sequence to the interests specified by the student. The optimal path for a student is computed dynamically and every action of the student can have an impact on the computed result. For example completing tests for a section makes it possible for the student to skip all sections and concepts contained in that section. Learners can start to work on the curriculum wherever they want. When a student selects a unit the system checks if a learner lacks any prerequisite knowledge to work on this unit and presents tests for lacking prerequisites. If a learner is not able to solve the given tasks, the system recommends to work on the prerequisites. Given any point in the curriculum, ACE can compute the next best unit depending on the probabilistic overlay model of a learner's knowledge and the prerequisites of possible next units.

In one experiment conducted the individual generation of an introductory lesson showed no significant effect on the knowledge gained. In tendency the results confirm that the prior knowledge of learners has a strong impact on the learning success with adaptive hypermedia [25]. Furthermore the effects of the explicit warning and recommendation of prerequisite units were confounded by the adaptive navigation support. To validate the results experimental studies on this single adaptive support should be done.

5.2.2. Dynamic adaptation of teaching strategy

Furthermore ACE generates views for learning units depending on the specified learner interests and learning material preferences. It computes the appropriate teaching strategy like described in section 2.2.1 and generates a personalized view on a learning unit like shown in figure 1. Basically, ACE has a default teaching strategy for each unit, which can be specified by the teacher (see Figure 3). In the introductory questionnaire, some information about the learners preferred learning style is captured and the system monitors which sequences of learning material are requested by a learner. For example, when a student often requests the introduction, looks for some examples, and then proceeds directly to the tests the system adapts to this learning strategy by presenting the links to the learning materials in this order. This is similar to approaches for adapting module sequences [33] or action sequences [34] in working environments to individual working styles. The adaptation to a preferred learning style takes into account the learner's success with the style. Strategies in which the learner shows a sufficient result in the final tests are rated as successful. Repeated occurrences raise the preference value of a style. Once a certain threshold is exceeded the learning strategy is taken as the default strategy for the learner.

Evaluations have shown that students that are following sequencing proposals adapted to their domain knowledge and interests gain a better understanding of the domain in shorter learning time [18].

5.3. Recommendations of a pedagogical agent

In ACE a pedagogical agent was implemented to give individualized recommendations to students dependent from their knowledge, their interests and their media preferences [18]. The implicitly given teaching goals were promoting and retaining learner motivation and evoking students' interest even in lessons they otherwise might have avoided.

The integrated agent "ADI" computed six different types of proposals. These proposals where computed by searching the knowledge base of the current curriculum with different constrains.

- a learning material of the current unit, corresponding with student's preferences. Following this proposal, the student may acquire the current unit by means of a type of learning material that he or she claimed to be especially convenient.
- a learning material of the current unit, to complete knowledge about this unit. This proposal draws the student's attention to a learning material of the current unit that has not yet been seen.
- a unit related to the current unit and corresponding with student's preferences. This proposition may attract the student's interest because it shows the idea of the current concept in the light of a global domain the student has claimed to be especially interested in.
- a unit to complete most recently acquired knowledge. The student's attention is drawn to another sub-unit within the current unit.
- a completely new, unrelated, unvisited chapter. A student, who is bored or annoyed by the section he or she is just working at, may like to follow this link.

- proceed with the current unit. In case, none of the above is considered convenient the student may simply continue where he or she interrupted to ask the agent for advice.

The overall results showed that students following the recommendations of the agent gained a better understanding, had more fun, and worked more intense [18].

Another feature attached to the agent was the visualization of the learner model. Learners could access a bar chart display of the current knowledge model in the system. From the bar chart they could directly jump to tests about a section to proof that they have better knowledge than assumed by the system. In the experiment done with ADI and the user model visualization the usage of the agent components were voluntary and the learners rarely used this feature. Therefore additional evaluations have to be done.

6. Summary and perspective

In this paper we sketched the architecture of the ACE framework for adaptable and adaptive learning environments on the WWW. Integrating a domain model, a learner model, and a pedagogical model allows several adaptive methods to be implemented. The adaptive methods implemented in the ACE framework are based on the pedagogical background of adaptive instruction and the psychology of learning. The basic ideas of this background are applied in the relatively new area of hypermedia courseware and are evaluated in this application field.

ACE courses have been built and evaluated for several domains and the adaptive methods applied showed to enhance learning compared to static hypertext learning environments. A main research goal for the future will be the empirical evaluation of adaptive methods in learning environments. These evaluations shall help to clarify the complex interaction effects of learner features and the characteristics of adaptive methods. The results could possibly show a way of how to adapt adaptive methods for different learners. All of the adaptive methods presented can easily be reused for general adaptive hypermedia applications where the main goal is not to learn. For such applications the adaptive methods could follow the same regularities but adapt to interests or professional background instead of knowledge state.

The ACE editor is just a first step towards an authoring environment for adaptive educational hypermedia. Further work will integrate a more elaborated library of diagnostic modules, teaching strategies, domain entities and relations, and will support more adaptive methods. The main goals for further developments are the simplicity of usage and the selection of an adequate level of adaptive parameters that can be specified by courseware authors.

7. References

1. Tennyson, R.D. and W. Rothen, Pre-task and On-task adaptive design strategies for selecting number of instances in concept acquisition. *Journal of Educational Psychology*, (1977). **69**: pp. 586-592.

2. Frank, H. *Lehrautomaten für die Einzel- und Gruppenschulung. Lehrmaschinen in kybernetischer und pädagogischer Sicht.* Nürnberg: Klett, 1965.

3. Tennyson, R.D. and D.L. Christensen, MAIS: An intelligent learning System. In: D.H. Jonassen, ed. *Instructional Designs for microcomputer courseware,* Hillsdale: N.J.:Erlbaum, 1988.

4. Pask, G., The adaptive teaching system. In: K. Austwick, ed. *Teaching machines and programming,* Oxford, London: Pergamon Press, 1964.

5. Leutner, D., *Adaptive Lehrsysteme. Instruktionspsychologische Grundlagen und experimentelle Analysen. Fortschritte der psychologischen Forschung,* Weinheim: Beltz, 1992.

6. André, E. R. T. and J. Müller. WebPersona: A Life-Like Presentation Agent for Educational Applications on the WWW. In: *8th World Conference of the AIED Society*. Kobe Japan, 1997.

7. Capell, P. and R.B. Dannenberg, Instructional design and intelligent tutoring: Theory and the precision of design. *Journal of Artificial Intelligence in Education.* **4(**1993), 1, 95 - 121.

8. Rickel, J. and L.W. Johnson, Integrating Paedagogical Agents in a virtual Environment for Training. *Journal Presence,* 1997.

9. Weber, G. and M. Specht. User modeling and Adaptive Navigation Support in WWW-Based Tutoring Systems. *Workshop on Adaptive Systems and User Modeling on the WWW on UM97*. Chia Laguna, Sardinia: Springer, 1997.

10. Brusilovsky, P., E. Schwarz, and G. Weber, A tool for developing hypermedia-based ITS on WWW. In: C. Frasson, G. Gauthier, and A. Lesgold, eds. *Intelligent Tutoring Systems - Proceedings of the Third International Conference*, ITS '96, Berlin: Springer, 1996, pp. 261-269.

11. London, R.V., Student Modeling to Support Multiple Instructional Approaches. *Journal of User-Modeling and User-Adapted Interaction*, 2(1991), 1-2, 117 - 154.

12. Feiner, S.K. and K.R. McKeown, Automating the Generation of Coordinated Multimedia Explanations. In: M.T. Maybury, ed. *Intelligent Multimedia Interfaces*, Boston: AAAI Press/MIT Press, 1993, pp. 117 - 138.

13. Goodman, B.A., Multimedia Explanations for Intelligent Training Systems. In: M.T. Maybury, ed. *Intelligent Multimedia Interfaces*, AAAI Press/MIT Press, 1993.

14. Carr, B. and I. Goldstein, *Overlays: A theory of modelling for computer aided instruction*. Cambridge, MA: Massachusetts Institute of Technology, AI Laboratory, 1977.

15. Weber, G., Episodic learner modeling. *Cognitive Science*, 20(1996), 2, 195 - 236.

16. Brusilovsky, P., Methods and techniques of adaptive hypermedia. *User Models and User Adapted Interaction*, 6(1996).

17. De Bra, P. and L. Calvi, AHA! An open Adaptive Hypermedia Architecture. *New Review of Hypermedia and Multimedia*, 1998. (this issue).

18. Schoech, V., M. Specht, and G. Weber. ADI — An Empirical Evaluation of a Pedagogical Agent. In: *World Conference on Educational Multimedia ED-MEDIA98*. Freiburg, Germany, 1998.

19. Specht, M., et al. AST: An adaptive WWW-Courseware for Statistics. in *Workshop on Adaptive Systems and User Modeling on the WWW on UM97*. 1997. Cia Lingua, Sardinia: Springer, 1997.

20. Ziep, K.-D., *Der Dozent in der Weiterbildung*, Weinheim: Deutscher Studien Verlag. 1990.

21. Wenger, E., *Artificial intelligence and tutoring systems. Computational approaches to the communication of knowledge.* Los Altos: Morgan Kaufmann. 1987.

22. Dannenberg, R.B., et al., A computer-based multi-media tutor for beginning piano students. *Interface*, 19(1990), 155 - 173.

23. Vassileva, J. A practical architecture for user modeling in a hypermedia-based information system. In: *4-th International Conference on User Modeling*. Hyannis: MITRE, 1994.

24. Nkambou, R. and G. Gauthier. Use of WWW resources by an intelligent tutoring system. In: *ED-MEDIA'96 - World conference on educational multimedia and hypermedia*. Charlottesville: AACE, 1996.

25. Specht, M., Adaptive Methoden in computerbasierten Lehr/Lernsystemen. Dissertation Psychology, GMD Forschungszentrum Informationstechnik, ed. GMD, *Research Series*. Vol. 1. (1998), St. Augustin.

26. Eklund, J., Knowledge-Based Navigation Support in Hypermedia Courseware using WEST. *Australian Educational Computing*, **11** (1996), (2).

27. Brusilovsky, P., E. Schwarz, and G. Weber. A tool for developing adaptive electronic textbooks on WWW. In: *WebNet'96, World Conference on the Web Society*. Charlottesville: AACE, 1996.

28. Eklund, J. and P. Brusilovsky. The Value of Adaptivity in Hypermedia Learning Environments: A Short Review of Empirical Evidence. In *The Ninth ACM Conference on Hypertext and Hypermedia*. Pittsburgh. Paper presented at The 2nd Workshop on Adaptive Hypertext and Hypermedia Held in Conjunction with HYPERTEXT '98, 1998.

29. Brusilovsky, P., Efficient Techniques of adaptive hypermedia. In: C. Nicholas and J. Mayfield, eds., *Intelligent Hypertext*, Springer: Berlin, Heidelberg, New York. 1997, pp. 12-27.

30. Specht, M. Empirical Evaluation of Adaptive Annotation in Hypermedia. in *ED-MEDIA98*. Freiburg Germany: AACE, Charlottesville 1998.

31. Specht, M. and A. Kobsa. Interaction of domain expertise and interface design. Paper presented at *Workshop on adaptive hypermedia held on conference UM99 and WWW99*. 1999.

32. Salomon, G., Heuristische Modelle für die Gewinnung von Interaktionshypothesen, In: *Adaptiver Unterricht*, R. Schwarzer, ed., Kösel: München. 1975, pp. 127-145.

33. Lindner, H.G. Ein Prototyp zur adaptiven Beratungsunterstützung des Versicherungsverkaufs. In: *Workshop Adaptivität und Benutzermodellierung in Interaktiven Softwaresystemen*. München: Siemens AG, 1995.

34. Maes, P., Modeling Adaptive Autonomous Agents. *Artificial Life Journal*, **1** (1994), 1.