Developing Adaptive Internet Based Courses with the Authoring System NetCoach

Gerhard Weber, Hans-Christian Kuhl, and Stephan Weibelzahl

Pedagogical University Freiburg, Germany
[weber,kuhl,weibelza]@ph-freiburg.de

Abstract. Developing adaptive internet based learning courses usually requires a lot of programming efforts to provide session management, keeping track of the learners current state, and adapting the interface layout to specific requirements. NetCoach is designed to enable authors to develop adaptive learning courses without programming knowledge. In this paper, we describe the adaptive, the adaptable, the interactive, and the communicative features of NetCoach. Both authors and tutors are supported in many ways to develop and manage courses via an online interface. Experiences with NetCoach courses in different domains and settings have shown that learners profit from the adaptive features.

1 Introduction

Internet based instruction has grown strongly during the last years. While in the beginning most internet based courses consisted only of a collection of static HTML-pages (mostly simple translations of already existing scripts and papers), a lot of sophisticated internet based learning systems emerged in recent time. The former systems could be easily created by authors using simple authoring tools, but these systems were not much more than copies of textbooks and lacked any adaptivity and guidance that would be needed to support learners when learning a new topic on their own. On the other hand, most current more sophisticated learning systems are proprietary solutions and can only be built by experienced programmers and skilled web-based instruction authors.

This puts high demands on authoring tools to create adaptive internet based instruction courses. In this paper, we will introduce NetCoach, an authoring system that meets the needs to create adaptive learning courses in the internet. Creating adaptive courses with NetCoach is very easy and can be done without being a skilled programmer. NetCoach is derived from ELM-ART[1], one of the first and by now most comprehensive adaptive web-based educational systems (Weber & Specht, 1997).

2 Features and Adaptivity in NetCoach-Courses

NetCoach is an authoring-system which allows to create adaptive and individual course modules without programming-knowledge. This section describes four characteristics that are common to all courses that have been developed with NetCoach. The Courses are adaptive, interactive, adaptable, and communicative.

[1] cogpsy.uni-trier.de/projects/ELM/elmart.html

2.1 Adaptive Elements in NetCoach Courses

According to Brusilovsky (1996), adaptive learning systems may adapt to the learners experience, knowledge, goals, or preferences. NetCoach adapts to the last three aspects of the user. This information can be used either to adapt the presentation of the content or to support the navigation (Brusilovsky, 1996). NetCoach implements two adaptive navigation techniques: *curriculum sequencing* and *adaptive annotation of links*. The goal of *curriculum sequencing* is to provide the student with the most suitable, individually planned sequence of knowledge units to learn and the sequence of learning tasks (examples, questions, problems, etc.) to work with. In other words, it helps the student to find an "optimal path" through the learning material. The goal of *adaptive annotation of links* is to support the student in hyperspace orientation and navigation by changing the appearance of visible links.

These adaptation techniques require a content specific knowledge-base and a user model that allows the system for responding individually to learners' interactions with the system.

The Knowledge-Base as Basis for Adaptive Behavior In NetCoach, the knowledge base of a course consists of concepts. These concepts are internal representations of pages that will be presented to the learner. In many domains the different concepts are related in many ways. To build up this knowledge-base, which is the basis for adaptive navigation support, the author can create many content-specific relations for every concept. However, the author is not forced to specify any relation. Default values that retain the sequential order of the concepts will be applied otherwise. Note, that specifying the concept relations is a simple procedure (as will be shown in Section 3) and is basically a content-specific task.

Normally the contents of a domain are related and interdependent. There are two relations between concepts: prerequisites and inferences.

First, the author can decide which other concepts are required to be learned to understand the current concept. These prerequisites can be chosen in the concept-list as shown in Figure 1 (*a*). The system will guide learners to theses prerequisite pages before suggesting the current concept. Because prerequisite concepts might have prerequisite concepts themselves there are also indirect prerequisites (*b*). In our example-course the system will recommend the following sequential order of concepts in case *Chapter-2-1-2* is the current learning goal: *Chapter-1* (indirect prerequisite), *Chapter 1-2*, *Chapter-2* (prerequisites), and finally *Chapter-2-1-2*.

Second, the inferences (*c*) of a concept are in some way the opposite of prerequisites. Perhaps an user wants to learn *Chapter-3-1* first and solves the test items correctly. Because *Chapter-2-1-2* is marked as inferred by *Chapter-3-1*, the system will assume that the user already knows *Chapter-2-1-2* as soon as *Chapter-3-1* has been worked at successfully. Note that prerequisites and inferences are related but not equal. E.g., knowing A might be required to understand B, but if one knows B this does not necessarily imply that A is known.

In addition to these relations between concepts the knowledge base contains relations between test items and concepts. Sets of test items (so called test groups) assess the user's current learning state of a concept (*d*). However, test items may not only test

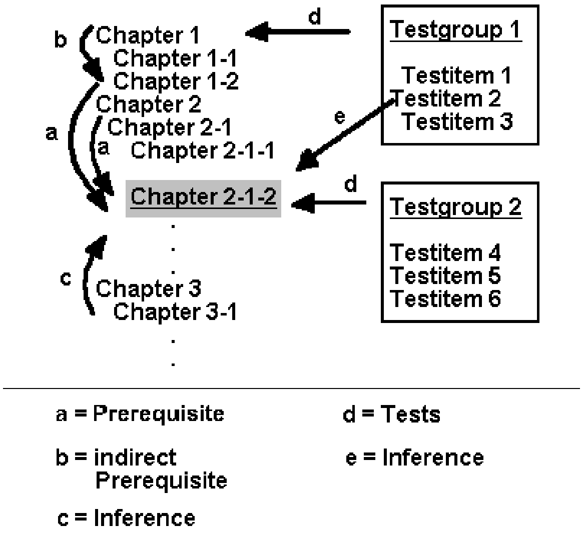


Fig. 1. Example of the relations of concept *Chapter-2-1-2*.

one concept but also assess aspects of other concepts. Thus, it is possible to quantify the inference of test items to other concepts. If the learner solves *testitem 2* in Figure 1 correctly (*e*), she has understood some important aspects of *Chapter-2-1-2*. A concept is supposed to be learned if one has reached a critical value. If there are already some inferences from test items of other concepts, the learner is closer to this critical value and has to solve less test items in *Testgroup 2* correctly.

The User Model. Based on the descriptions in the concepts, all pages are computed individually with respect to the learner's user model. The user model used in NetCoach is a multi-layered overlay model (Weber, 1999). Individual information about each learner is stored with respect to the concepts of the course's knowledge base (as described in the previous section). The first layer describes whether the user has already visited a page corresponding to a concept. The second layer contains information on which exercises or test items related to this particular concept the user has worked at and whether he or she has successfully worked on the test items up to a certain criterion. The third layer describes whether a concept could be inferred as known via inference links from more advanced concepts the user has already worked on successfully. Finally, the fourth layer describes whether a user has marked a concept as already known. That is, the user model can be inspected and edited (Bull & Pain, 1995). Sometimes, this is called a cooperative user model (Kay, 1995). Information in the different layers is updated independently. This leads to the fact that information from each different source does not overwrite others. E.g., if a student unmarks a concept because she realized that she has not enough pre-knowledge about it, the information about tests on this concept is still available.

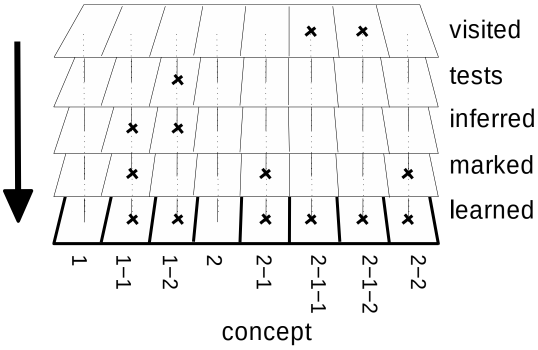
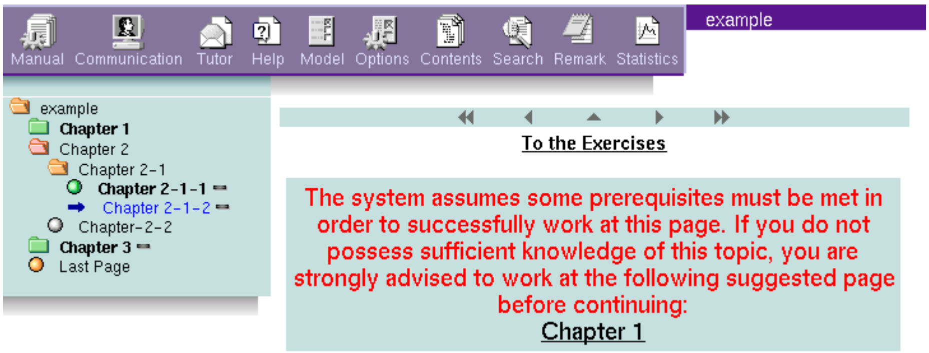


Fig. 2. Example of a student's overlay model. NetCoach infers the student's current learning state from four independently updated layers. Concepts without tests are treated as learned if they have been visited.

See Figure 2 for an example of a student's overlay model. A concept is assumed to be learned if it is either tested to be known, inferred from other learned concepts, or marked by the user. In case no test group is available the concept is assumed to be learned if it has been visited. I.e., the visited layer and the test layer are applied alternatively.

Curriculum Sequencing and Link Annotation. The multi-layered overlay model supports both the adaptive annotation of links and individual curriculum sequencing. Links that are shown in an overview on each page or in the table of contents are visually annotated in correspondence to the user's current learning state. Individual curriculum sequencing means that the system's suggestion which page is best to be visited next is computed dynamically according to the general learning goal and the user's learning state of the concepts. Users get a warning if they visit a page with missing prerequisites. However, access to that page is not restricted and the warnings can be turned off. See Figure 3 for an example of a warning due to unfulfilled prerequisites, the corresponding page suggestion, and the link annotation in the overview frame on the lefthand side.

Learning Goals. In addition, NetCoach supports the specification of learning goals. A goal consists of a set of concepts that have to be successfully worked on by the learner. All (direct and indirect) prerequisites are computed automatically and corresponding pages are suggested. Thus, learning goals are especially useful for learners that do not want to complete the whole course. E.g., the goal "I want to get an introduction on this topic" might include the introductory chapters only, while the second goal "I am familiar with ..., but I want to know more about ..." would leave out the first chapter and suggest to go to the advanced sections directly.



The system assumes some prerequisites must be met in order to successfully work at this page. If you do not possess sufficient knowledge of this topic, you are strongly advised to work at the following suggested page before continuing:
Chapter 1

Fig. 3. Screenshot of the adaptive learning environment including curriculum sequencing and link annotation.

2.2 Interactive Elements in NetCoach Courses

Online-presented and evaluated exercises and tests are central features of interactive courses. NetCoach provides the possibility to present exercises and tests in different formats. These are multiple choice, forced choice, gap filling tests, open questions and e-mail-questions. While the e-mail-questions will be evaluated individually by human tutors, open questions have an example-answer as feedback, so that the learners can compare their solution by themselves. The feedback for the remaining item formats consists of a hint which answer is correct and an explanation why the answer was false or correct. Moreover, it is possible to give at first a hint only, before the correct answer is given.

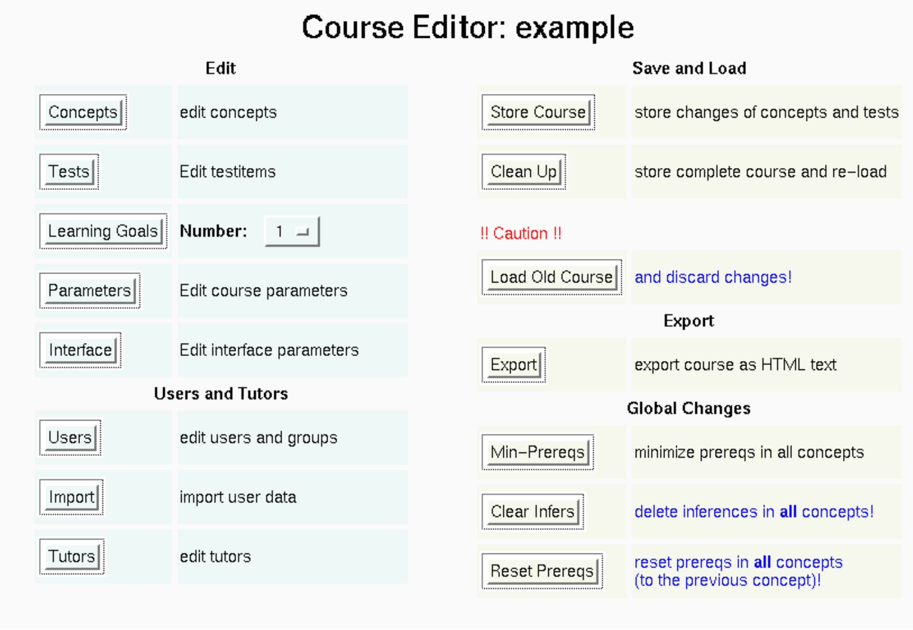
NetCoach Courses can be additionally shaped highly interactive by connecting animations (e.g., flash-animations). These interactive animations can be contained like in every normal web-page. Animations can provide interactive work in simulated scenarios with multiple interactive mouse-events.

A glossary and a page with references can be accessed by the users with direct links in the text or a button. Finally, a search-tool and a notice-board are available.

2.3 Adaptable Elements in NetCoach Courses

Web-based courses are used by users with very different knowledge and different computer skills. Because of that it is useful if learners can adapt the learning environment to their own needs.

Not only the developed courses adapt to the user, but also the users themselves can adjust many features for their own preferences. Especially the kind of presentation, warnings and recommendations can be changed or switched off. The model behind the adaptive functions is not incomprehensible but can be investigated and changed by the learners themselves. The manner of annotation and the feedback can be adjusted, too.



Course Editor: example

Edit	
Concepts	edit concepts
Tests	Edit testitems
Learning Goals	**Number:** 1
Parameters	Edit course parameters
Interface	Edit interface parameters
Users and Tutors	
Users	edit users and groups
Import	import user data
Tutors	edit tutors

Save and Load	
Store Course	store changes of concepts and tests
Clean Up	store complete course and re-load
!! Caution !!	
Load Old Course	and discard changes!
Export	
Export	export course as HTML text
Global Changes	
Min-Prereqs	minimize prereqs in all concepts
Clear Infers	delete inferences in **all** concepts!
Reset Prereqs	reset prereqs in **all** concepts (to the previous concept)!

Fig. 4. Screenshot of the online-interface for authors

2.4 Communicative Elements in NetCoach Courses

Courses developed with NetCoach provide different syncronous and asyncronous communication tools. Questions, proposals etc. can be sent via e-mail to human tutors. A chat module provides direct communication between students. Besides, there is a possibility to discuss the contents in different discussion lists where the learners can exchange opinions and ask questions. It is also possible for every learner to exchange documents (e.g., word documents or pdf) with other learners.

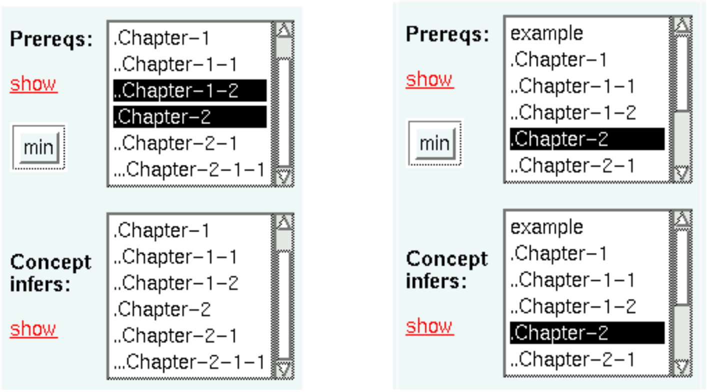
All these communicative features enable lectures and teachers to organize complete virtual courses where students can interact with each other, but are still free to learn at their individual speed.

3 The NetCoach Authoring System

The NetCoach authoring-system bases on a LISP-server (CL-HTTP[2]) / web-browser-client technology. NetCoach[3] is available for Windows, Apple, and Linux operating-systems. Learners, tutors and even authors just need a standard-web-browser to work with the corresponding interfaces.

The goal of developing NetCoach was to provide authors with a tool to create highly adaptive courses without being required to program user models or interactive tests.

[2] www.ai.mit.edu/projects/iiip/doc/cl-http/home-page.html
[3] www.net-coach.de



Prerequisites and inferences of *Chapter-2-1-2*

Prerequisites and inferences of *Chapter-3-1*

Fig. 5. Screenshot of a the author interface for specifying the prerequisites and the inferences of the concepts *Chapter-2-1-2* and *Chapter-3-1*.

The NetCoach authoring tool supports the complete developing process of adaptive web-based courses which includes authoring the learning material (e.g., texts, pictures), composing tests, defining learning goals, and adapting the layout and behavior of the interface. Figure 4 shows a screenshot of NetCoach's online-interface for authors.

In the concept editor, the concepts of the learning course (corresponding to pages presented in a browser) are described. Concepts may be arranged hierarchically similar to chapters and subchapters in a book. Authors may simply type in plain text, paste code from a HTML-editor, or even import an already existing HTML-file. In addition, animations that are created with Java, JavaScript or common plug-ins (e.g., flash animations) are supported as well.

The prerequisites on a concept and the inferences that can be drawn from successfully learning a concept are described by selecting the corresponding concepts from a table. See Figure 5 for a screenshot of how an author specifies the prerequisites and inferences of the concept *Chapter-2-1* which has been described in Section 2.1. To achieve perfect adaptivity effects authors have to define these concept relations very carefully in dependence of the domain structure.

In the test editor, test items can be defined and tested. The test editor offers templates for all test types, so authors are not required to program complicated cgi- or Java-scripts. In fact, NetCoach presents the test-questions, evaluates the answers, and observes the learning state of each user automatically, while the author can focus on the contents of the test items. Each item consists of three parts: First, the question that will be presented to the user. Second, the correct answers that have to be filled in by the user (gap filling, free input) or that have to be marked (forced choice, multiple choice). Third, authors can provide an explanation for the solution to help learners in understanding why they were wrong. Some of our courses include item pools of more than thousand items.

Test items are collected in test groups that are assigned to concepts. These test groups can be used as exercises or as introductory and final questionnaires. The learner's success on working on these test items is used to compute the user's learning state in the multi-layered learner model as described above.

Finally, a course might implement different learning goals. A learning goal consists of a set of concepts that has to be completed. Learners who decide not to work on the complete course but to fulfill a subgoal will receive individually recommendations which concept to visit next to complete this goal. The author has just to specify the concepts that are necessary to complete a goal. The prerequisites are computed automatically based on the hypertext model.

NetCoach not only is highly flexible in presenting different contents but also in adapting the course layout. Many optional parameters specify which buttons are presented how and where, which services are available (e.g., communication, search, or manual), and which components are adaptable by the user. This flexibility makes it easy to meet the requirements of different settings and even experimental studies. Moreover, NetCoach supports multiple languages, so courses can be developed in different target languages at the same time.

The NetCoach editors work in direct interaction with the NetCoach server so that it is possible to see effects of changing parameter settings, concepts, or test items directly in the course under development. This makes the creation of very sophisticated courses easy without requiring programming knowledge. A short tutorial on creating courses with NetCoach is described at http://art.ph-freiburg.de/NetCoach-Tutorial.

4 Tutors in NetCoach Courses

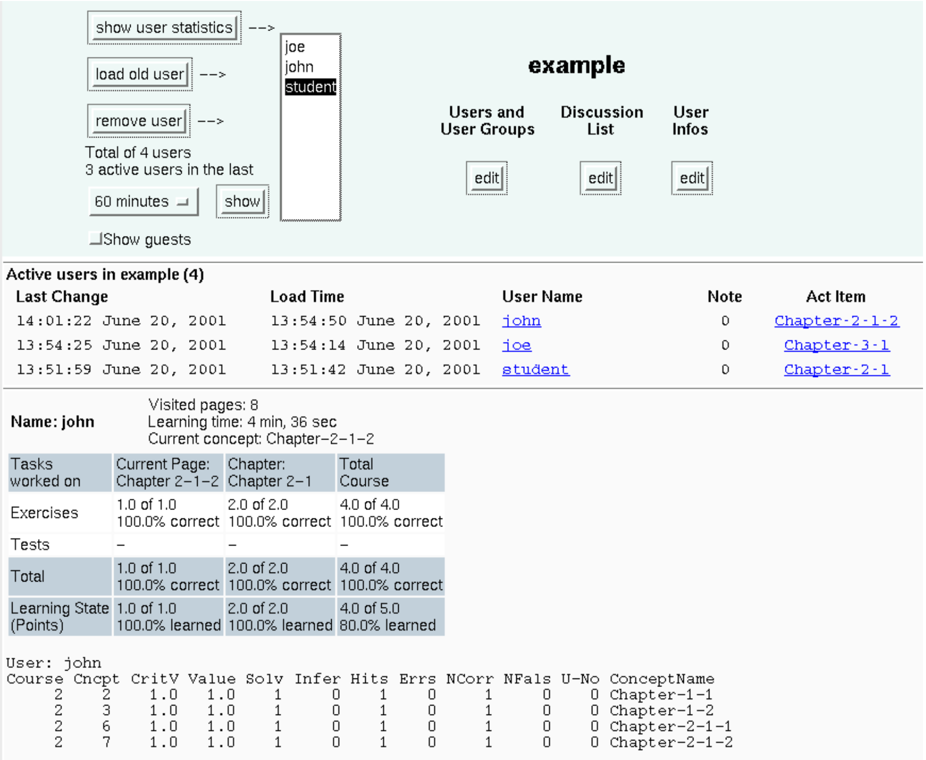
Courses created with NetCoach are guided by tutors that aid users on help requests, inspect user data, edit discussion lists, send messages to users, and manage user accounts and user groups. This is done via an online-interface which is shown in Figure 6.

Authors of a course can register tutors via the main course editor. Tutors have their own access rights and are able to inspect the course and users in the course.

First, tutors can get into contact with users. In a tutor help window, users can ask questions to tutors or give remarks on the course. The text directly typed into the tutor help window is sent by the server to the tutor (or tutors) by e-mail. Tutors can respond by e-mail in case the user has provided his or her e-mail address or send a message that is stored with the learner's user model and will be displayed to the user with the next page the user opens in the course.

Second, tutors can observe users in the course. They get a list of all currently active users and have access to all users in the course. Tutors can inspect the current learning state of a user. That is, they can see how long users have been working at the course, which messages they have sent to tutors, which concepts they have worked at, how many errors they have made, and some more interesting information that may help a tutor to understand the difficulties learners have with the course in order to help these learners.

Third, tutors can manage a discussion list. They can provide new topics in the discussion list, change trees of contributions to the discussion list to a new topic, and



Total of 4 users
3 active users in the last

Active users in example (4)

Last Change	Load Time	User Name	Note	Act Item
14:01:22 June 20, 2001	13:54:50 June 20, 2001	john	0	Chapter-2-1-2
13:54:25 June 20, 2001	13:54:14 June 20, 2001	joe	0	Chapter-3-1
13:51:59 June 20, 2001	13:51:42 June 20, 2001	student	0	Chapter-2-1

Name: john

Visited pages: 8
Learning time: 4 min, 36 sec
Current concept: Chapter–2–1–2

Tasks worked on	Current Page: Chapter 2–1–2	Chapter: Chapter 2–1	Total Course
Exercises	1.0 of 1.0 100.0% correct	2.0 of 2.0 100.0% correct	4.0 of 4.0 100.0% correct
Tests	–	–	–
Total	1.0 of 1.0 100.0% correct	2.0 of 2.0 100.0% correct	4.0 of 4.0 100.0% correct
Learning State (Points)	1.0 of 1.0 100.0% learned	2.0 of 2.0 100.0% learned	4.0 of 5.0 80.0% learned

```
User: john
Course Cncpt CritV Value Solv Infer Hits Errs NCorr NFals U-No ConceptName
     2     2   1.0   1.0    1     0    1    0     1     0    0 Chapter-1-1
     2     3   1.0   1.0    1     0    1    0     1     0    0 Chapter-1-2
     2     6   1.0   1.0    1     0    1    0     1     0    0 Chapter-2-1-1
     2     7   1.0   1.0    1     0    1    0     1     0    0 Chapter-2-1-2
```

Fig. 6. Screenshot of the online-interface for tutors

remove parts of the discussion list. In the discussion lists, learners can place statements that can be read by all other learners from the same course or make remarks on those statements. Tutors can watch these discussion lists and remove contributions that are not related to the topics of the course.

Fourth, tutors can put information messages to all users of a course or to members of a user group. These messages have an expiration date so that messages that remind of an important date, for example, will be removed automatically after expiration. These information messages will be displayed to a user when he or she (re-)enters a course or even during working at the course.

Fifth, tutors have to manage user accounts and user groups. In case of closed courses (that is, the courses are restricted to users that have access rights), tutors can add new users or remove users, give access rights to users and assign users to working groups. Users assigned to working groups can exchange documents via the server. Additionally, users communicate directly with other members of the group via e-mail or chat.

5 Comparison with other Authoring Systems

5.1 General Features of Authoring Systems

Presently there are many authoring systems to create virtual learning environments. Most of them are non-adaptive or just adaptable like WebCT. In the following, NetCoach will be compared to the well distributed authoring systems WebCT (Goldberg, Salari, & Swoboda, 1996), Learning-Space[4], and TopClass[5]. These three authoring-systems where chosen because they are also based on a server/Web-Browser-Client technology and are already in use. In Section 5.2 we will compare NetCoach to other authoring systems that are designed to deliver adaptive hypermedia courses.

WebCT can be used flexibly to create entire online courses, or to publish materials that supplement existing courses. All interaction with WebCT takes place through a web browser. Essentially a WebCT course consists of a series of linked HTML pages that define a path or "road-map" through the course material. The course content is supplemented by WebCT tools which can be built into the course design by simply dragging the appropriate tool icon onto the web page.

TopClass courses are constructed of Units of Learning Material (ULMs). These ULMs can consist of pages, exercises, or further ULMs themselves. ULMs can be freely exported and imported from course to course. In addition to course management, TopClass also manages student progress, user-tracking, and access to course materials.

Learning Space is based on Lotus Notes and uses Notes Server technology to provide a secure environment with a rich set of tools. Learning space includes tools for browsing the web and inserting multimedia material into learning space documents. Links can be defined from Learning Space to multimedia content on the web. Additionally resources and other content may be exchanged via the Media Center. Completed courses may be archived by the instructor for future use. A Portfolio is contained in every participant's Profile. This is a secure area for returned assignments and assessments which can only be viewed by the participant and the tutor.

As shown in Table 1 there is much conformity comparing NetCoach and the other authoring-systems described above regarding the functionality for authors, tutors and learners. Authors can import contents or contain multimedia elements. Tutors can investigate, add or delete user data and provide discussion lists. Learners can use different asynchronous and synchronous communication tools like e-mail, discussion-lists, file-exchange, and chat. Whitebord, video conferences and homepage-authoring are not implemented in all systems and as well not in NetCoach. A web browsing tool only exists in Learning Space.

The main differences are the adaptive possibilities in course-development with NetCoach. These features like curriculum-sequencing and dynamic link-annotation are described in section 2.1. The course-management (e.g. registration, examinations, calendar) which is more central in the other systems is less important in NetCoach. NetCoach is mainly a system to develop entire, adaptive courses. For this reason, NetCoach will be compared with other intelligent systems in the following section.

[4] www.lotus.com

[5] www.wbtsystems.com

Table 1. Comparison of authoring systems for web-based training

	NetCoach	**WebCT**	**TopClass**	**Learning Space**
Author Tools				
adaptive guiding	yes			
adaptive link annotation	yes			
creating / importing content	yes	yes	yes	yes
add / play multimedia content	yes	yes	yes	yes
Tutor Tools				
store & view learner data	yes	yes	yes	yes
add / remove learners	yes	yes	yes	yes
performing assessments	yes	yes	yes	yes
create discussion groups	yes	yes	yes	yes
Student Tools				
adaptable preferences	yes	by tutor	yes	Yes
web browsing				Yes
creating / importing content		yes		Yes
store bookmarks		yes	yes	Yes
play multimedia	yes	yes	yes	Yes
homepage authoring		yes	yes	Yes
calendar tool		yes		Yes
searchable resource archive		yes		Yes
Communication				
e-mail	yes	yes	yes	Yes
noticeboard	yes	yes	yes	Yes
file exchange	yes	yes	yes	Yes
asynchronous discussions	yes	yes	yes	Yes
chat	yes	yes	Add. module	Yes
whiteboard		yes		Yes
video conferencing			Add. module	Yes
Technology				
server/web-browser-client	yes	yes	yes	Yes

Table 2. Comparison of authoring systems for adaptive hypermedia courses in reference to the user's features that the systems adapt to and the methods that are used for adaptation.

	Net-Coach	AHA	ECSAI-Web	Inter-book	Meta-Links
user features: to what?					
goals	yes		yes		yes
navigation history	yes	yes	yes	yes	yes
tested knowledge	yes		yes		
preferences	yes		yes		yes
methods: how?					
adaptive guidance	yes	yes	yes	yes	yes
adaptive annotation	yes	yes	yes	yes	yes
adaptive hiding of links		yes			
adaptive navigation maps					yes
adaptive text presentation		yes			yes

5.2 Adaptivity Features of Authoring Systems

Several other authoring systems aim at delivering adaptive web-based courses. We selected four of them to highlight the strengths and weaknesses of adaptivity in NetCoach: AHA (De Bra & Calvi, 1998), ECSAIWeb (Sanrach & Grandbastien, 2000), Interbook (Brusilovsky, Eklund, & Schwarz, 1998), and MetaLinks (Murray, Shen, Piemonte, Condit, & Thibedeau, 2000).

See Table 2 for a comparison of the user's features that the systems adapt to and the methods that are used for adaptation. The comparison is based on the categorization of adaptive hypermedia systems introduced by Brusilovsky (1996).

NetCoach implements most commonly used adaptive features, but does not adapt the text presentation (as e.g., AHA) and refrains from hiding links. We argue that the student should have full freedom of navigation and content access while the adaptive system should provide hints and suggestions only. However, NetCoach's adaptations are based on a diversity of user data. Especially knowledge assessment with tests is only found in ECSAIWeb and NetCoach. The overlay model in ECSAIWeb is slightly less sophisticated as it does currently not consider the inference relation (section 2.1) for adaptation purposes.

6 Conclusion

Several courses have been developed with NetCoach. They are used at different universities in Germany and in some companies. Up to now, most courses are written in German, though some are written in English (ELM-ART) or in French. Because NetCoach does not require any programming knowledge, many different authors from many disciplines developed courses in different domains including programming, spelling rules, cognitive and pedagogical psychology, and product presentation. At the Pedagogical

University in Freiburg, students develop simple courses on their own and test these courses with pupils in secondary schools. NetCoach has been used for “learning on demand” settings, as well as for supplementing courses at universities and adult education. E.g., several courses on pedagogical psychology are used by students to prepare lessons and exams, while two courses on programming LISP and HTML are available world-wide for training purposes. Accordingly, the courses differ widely in structure and features to suit the specific settings. Experiences with these courses show that users can learn easily and successfully. Results of several investigations support the usefulness of the adaptive features of NetCoach (Weber & Specht, 1997; Weibelzahl, 2001).

References

Brusilovsky, P. (1996). Methods and techniques for adaptive hypermedia. *User Modeling and User-Adapted Interaction*, *6*(2-3), 87–129.

Brusilovsky, P., Eklund, J., & Schwarz, E. (1998). Web-based education for all: A tool for developing adaptive courseware. In *Computer Networks and ISDN Systems. Proceedings of Seventh International World Wide Web Conference, 14-18 April 1998* (Vol. 30, pp. 291–300).

Bull, S., & Pain, H. (1995). “Did I say what I think I said, and do you agree with me?” Inspecting and questioning the student model. In J. Greer (Ed.), *Artificial Intelligence in Education, Proceedings of AI-ED’95, 7th World Conference on Artificial Intelligence in Education, 16-19 August 1995. Washington, DC, AACE* (pp. 501–508).

De Bra, P., & Calvi, L. (1998). AHA! An open adaptive hypermedia architecture. *The New Review of Hypermedia and Multimedia*, *4*, 115–139.

Goldberg, M. W., Salari, S., & Swoboda, P. (1996). World Wide Web - course tool: An environment for building WWW-based courses. *Journal of Computer Networks and ISDN Systems*, *28*(7-11), 1219–1231.

Kay, J. (1995). The UM toolkit for cooperative user models. *User Models and User Adapted Interaction*, *4*(3), 149–196.

Murray, T., Shen, T., Piemonte, J., Condit, C., & Thibedeau, J. (2000). Adaptivity in the MetaLinks hyper-book authoring framework. In *Workshop Proceedings of Adaptive and Intelligent Web-Based Education Systems workshop at ITS 2000.*

Sanrach, C., & Grandbastien, M. (2000). ECSAIWeb: A web-based authoring system to create adaptive learning systems. In P. Brusilovsky, O. Stock, & C. Strapparava (Eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems. International Conference, AH 2000* (pp. 214–226). Berlin: Springer.

Weber, G. (1999). Adaptive learning systems in the World Wide Web. In J. Kay (Ed.), *User modeling: Proceedings of the Seventh International Conference, UM99* (pp. 371–378). Vienna: Springer.

Weber, G., & Specht, M. (1997). User modeling and adaptive navigation support in WWW-based tutoring systems. In A. Jameson & C. Tasso (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97* (pp. 289–300). Vienna: Springer.

Weibelzahl, S. (2001). Evaluation of adaptive systems. In M. Bauer (Ed.), *User modeling: Proceedings of the Eighth International Conference, UM2001.*