PETER BRUSILOVSKY

DEVELOPING ADAPTIVE EDUCATIONAL HYPERMEDIA SYSTEMS: FROM DESIGN MODELS TO AUTHORING TOOLS

School of Information Sciences
University of Pittsburgh
Pittsburgh PA 15260, USA
peterb@mail.sis.pitt.edu

Abstract. Adaptive hypermedia is a new area of research at the crossroads of hypermedia, adaptive systems, and intelligent tutoring systems. Educational hypermedia systems is cirrently the most popular kind of adaptive hypermedia. The goal of this paper is to uncover the secrets of authoring adaptive educational hypermedia. The paper provides a clear structured view on the process of adaptive hypermedia authoring starting from the early design stage. It also reviews a few modern adaptive hypermedia authoring systems that are oriented to educational practitioners.

1. INTRODUCTION

Adaptive hypermedia (AH) is an alternative to the traditional “one-size-fits-all” approach in the development of hypermedia systems. Adaptive hypermedia systems (AHS) build a model of the goals, preferences and knowledge of each individual user, and use this model throughout the interaction with the user, in order to adapt to the needs of that user (Brusilovsky, 1996). For example, a student in an adaptive educational hypermedia system will be given a presentation that is adapted specifically to his or her knowledge of the subject (De Bra, & Calvi, 1998), and a suggested set of most relevant links to proceed further (Brusilovsky, Eklund, & Schwarz, 1998b). An adaptive electronic encyclopedia will personalize the content of an article to augment the user's existing knowledge and interests (Milosavljevic, 1997). A virtual museum will adapt the presentation of every visited object to the user's individual path through the museum (Oberlander, O'Donell, Mellish, & Knott, 1998).

AH systems can be useful in any application area where users of a hypermedia system have essentially different goals and knowledge and where the hyperspace is reasonably large. Users with different goals and knowledge may be interested in different pieces of information presented on a hypermedia page and may use different links for navigation. AH tries to overcome this problem by using knowledge represented in the user model to adapt the information and links being presented to the given user. Adaptation can also assist the user in a navigational sense, which is particularly relevant for a large hyperspace. Knowing user goals and knowledge, AH systems can support users in their navigation by limiting browsing space, suggesting most relevant links to follow, or providing adaptive comments to visible links.

It is quite natural that educational hypermedia was one of the first application areas for AH. In educational context users with alternative learning goals and knowledge on the subjects require essentially different treatment. It is also in educational hypermedia where the problem of "being lost in hyperspace" is especially critical. A number of pioneer adaptive educational hypermedia systems were developed between 1990 and 1996. These systems can be roughly divided into two research streams. The systems of one of these streams were created by researchers in the area of intelligent tutoring systems (ITS) who were trying to extend traditional student modeling and adaptation approaches developed in this field to ITS with hypermedia components (Beaumont, 1994; Brusilovsky, Pesin, & Zyryanov, 1993; Gonschorek, & Herzog, 1995; Pérez, Gutiérrez, & Lopistéguy, 1995). The systems of another stream were developed by researchers working on educational hypermedia in an attempt to make their systems adapt to individual students (De Bra, 1996; de La Passardiere, & Dufresne, 1992; Hohl, Böcker, & Gunzenhäuser, 1996; Kay, & Kummerfeld, 1994).

Table 1. Design and authoring steps in the process of creating regular and adaptive educational hypermedia systems. The steps that are not essential for the whole process and can be skipped in special cases are shown in italics.

Regular educational hypermedia	*Adaptive educational hypermedia*
Design	
	Design and structure the knowledge space
	Design a generic user model
	Design a set of learning goals
Design and structure the hyperspace of educational material	Design and structure the hyperspace of educational material
	Design connections between the knowledge space and the hyperspace of educational material
Authoring	
Create page content	Create page content
Define links between pages	Define links between pages
	Create some description of each knowledge element
	Define links between knowledge elements
	Define links between knowledge elements and pages with educational material

The first adaptive educational hypermedia systems were relatively small with the hyperspace rarely exceeding 100 nodes. The focus of adaptive hypermedia research during the first 4-6 years was in developing innovative user modeling and adaptation technologies (Brusilovsky, 1996). As long as this field was moving to a more mature state with a good number of established and evaluated adaptation technologies, the researchers in the field of adaptive hypermedia attempted to build the systems with incrementally larger hyperspace. This trend was caused by both, availability of ready to be used technologies and by the growing needs of Web-based education. As a result, during the last 5 years the focus of research has gradually moved from creating more and more new AH technologies to the problems of design and authoring of AH systems. A number of design frameworks suitable for building large adaptive hypermedia systems were suggested (Brusilovsky, 1997; Grigoriadou, Papanikolaou, Kornilakis, & Magoulas, 2001; Neumann, & Zirvas, 1998; Steinacker, Seeberg, Rechenberger, Fischer, & Steinmetz, 1999; Stern, & Woolf, 2000; Süß, Kammerl, & Freitag, 2000) and a few authoring toolkits and systems were developed (Brusilovsky, et al., 1998b; De Bra, et al., 1998; Forcheri, Molfino, Moretti, & Quarati, 2001; Hockemeyer, Held, & Albert, 1998; Murray, 2001; Sanrach, & Grandbastien, 2000; Specht, & Oppermann, 1998; Weber, Kuhl, & Weibelzahl, 2001b).

This paper provides a summative review of the state of the art in adaptive hypermedia authoring. Instead of analyzing existing tools and frameworks one by one, I have attempted to provide a coherent unified view to the various aspects of AH authoring. The goal of this review is to help the potential authors of adaptive hypermedia systems to understand the variety of opportunities in this field and to select a framework or a tool that is most relevant to the needs of their subjects.

2. THE STRUCTURE OF THE REVIEW

The secret of adaptivity in all adaptive hypermedia systems is "knowledge behind pages". All adaptive educational hypermedia systems explicitly model the knowledge of the domain to be taught in the form of elementary knowledge elements or concepts that form a knowledge space. To let the adaptive system "know" what is presented on a particular page or page fragment author of an AH system has to specify the knowledge elements "behind" it thus creating links between the knowledge space and the hyperspace of educational material. As a result, the design and authoring of educational AHS is more complicated than the design of regular educational hypermedia. In addition to structuring the hyperspace and authoring the pages with educational content, the author of an adaptive hypermedia system need also to structure the knowledge space and define the connections between knowledge space and hyperspace of educational material. Table 1 shows the list of design and authoring stages for regular and adaptive educational hypermedia. As we see, there are a number of additional steps that have to be performed when creating adaptive educational hypermedia systems. Even though some steps (shown in italics) are not are not essential and can be skipped in some simpler systems, the job of creating an AH system is considerably harder than the job of creating a traditional hypermedia system with the same size of the hyperspace. This is the price of adaptivity.

I have structured this review according to the sequence of design and authoring steps to be performed when creating adaptive educational hypermedia systems. Two major parts of this paper correspond to two major stages – design and authoring. The subsections of the design part correspond to design and authoring steps. Each subsection presents a "universal approach" for this step and discusses known variations of this approach. The subsections in the authoring part correspond to major authoring approaches and explain how each of these approaches support main authoring steps. At the end I attempt to predict the future developments in the area of adaptive educational hypermedia authoring and clarifies his viewpoint on the design of adaptive hypermedia systems presented in this paper.

3. DESIGN STAGE: STRUCTURING THE INFORMATION

The information structure of a typical adaptive hypermedia system can be considered as two interconnected networks or "spaces" (Figure 1): a network of concepts (knowledge space) and a network of hypertext pages with educational material (traditional hyperspace). Accordingly, the design of an adaptive hypermedia system involves three key sub-steps: structuring the knowledge, structuring the hyperspace, and connecting the knowledge space and the hyperspace. On each step a developer of an AH system can choose from several known ways that are reviewed below. Also, in this part I review student model and learning goals design steps since they are to a large extent defined by a set of design choice made in structuring the domain information.

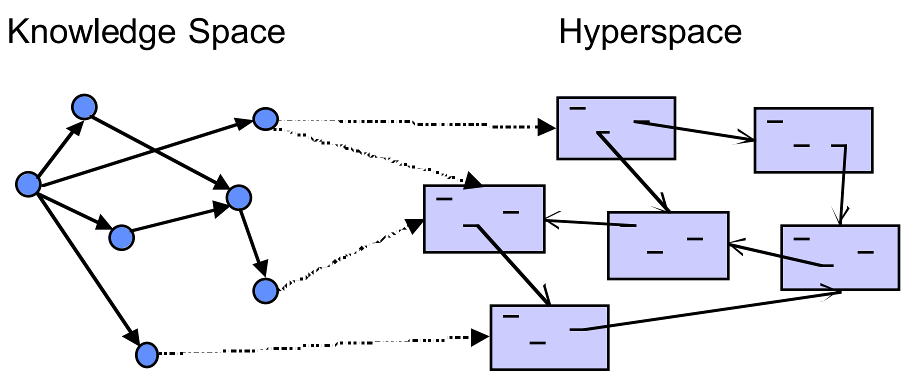


Figure 1. A typical structure of information space in an adaptive hypermedia system

3.1. Structuring the Knowledge

3.1.1. The Domain Model

The heart of the knowledge-based approach to developing adaptive hypermedia systems is a structured *domain model* that is composed of a set of small domain knowledge elements (DKE). Each DKE represents an elementary fragment of knowledge for the given domain. DKE concepts can be named differently in different systems—concepts, knowledge items, topics, knowledge elements, learning objectives, learning outcomes, but in all the cases they denote elementary fragments of domain knowledge. In this paper I will be calling DKE as *concepts*.

Though this name is slightly misleading[1] it is currently the most popular way to name DKE. Depending on the domain, the application area, and the choice of the designer concepts can represent bigger or smaller pieces of domain knowledge. A set of domain concepts forms *a domain model.* More exactly, a set of independent concepts is the simplest form of domain model. I call it a *set model* or a *vector model* since the set of concepts has no internal structure. In a more advanced form of domain model concepts are related to each other thus forming a kind of semantic network. This network represents the structure of the domain covered by a hypermedia system. I call this kind of model a *network model* (Figure 2).

The structured domain model was inherited by adaptive educational hypermedia systems from the field of ITS where it was used mainly by systems with task sequencing, curriculum sequencing, and instructional planning functionality (Brusilovsky, 1992). The reason for that is that most of the early adaptive educational hypermedia systems had strong connection with ITS systems. In fact, a number of them were developed in an attempt to extend an ITS system with hypertext functionality (Beaumont, 1994; Brusilovsky, Schwarz, & Weber, 1996a; Brusilovsky, 1993; Gonschorek, et al., 1995; Pérez, et al., 1995). This model proved to be relatively simple and powerful and was later accepted as de-facto standard by almost all educational and many non-educational adaptive hypermedia systems.

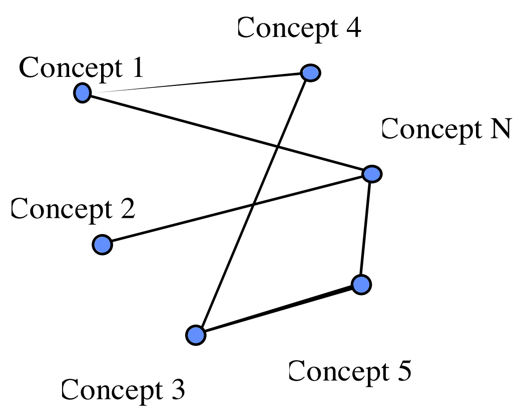


Figure 2. A network domain model

Domain models in adaptive hypermedia systems seriously differ in complexity. It could be surprising that some AHSs developed for teaching practical university courses employed simplest vector domain model (Brusilovsky, & Anderson, 1998a; De Bra, 1996). At the same time, several modern AHSs use networked models with several kinds of links that represent different kinds of relationships between concepts. The most popular kind of links in educational AHS is prerequisite links between concepts which represent the fact that one of the related concepts has to be learned before another. Prerequisite links are relatively easy to understand by authors of AHS and can support several adaptation and user

1 The name concepts can cause someone to think that concepts can only represent fragments of conceptual knowledge. However, a concept is a general name that can denote a fragment of knowledge of any kind, including procedural knowledge.

modeling techniques. In many AHS prerequisite links is the only kind of links between concepts (Grigoriadou, et al., 2001; Henze, Naceur, Nejdl, & Wolpers, 1999a; Hockemeyer, et al., 1998; Neumann, et al., 1998; Pilar da Silva, Durm, Duval, & Olivié, 1998; Weber, et al., 2001b). Other kinds of links that are popular in many systems are classic semantic links "is-a" and "part-of" (De Bra, & Ruiter, 2001; Steinacker, et al., 1999; Vassileva, 1998).

Another difference in complexity is related with internal structure of concepts. For the majority of educational AHS the domain concepts are nothing more than names that denote fragments of domain knowledge. At the same time, some AH systems use a more advanced frame-like knowledge representation, i.e., represent an internal structure of each concept as a set of attributes (Beaumont, 1994; Hohl, et al., 1996; Weber, & Brusilovsky, 2001a).

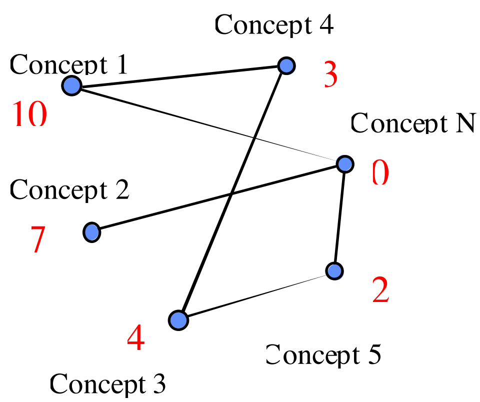


Figure 3. A simple overlay student model

3.1.2. The student model

One of the most important functions of the domain model is to provide a framework for representation of the user's domain knowledge. The majority of educational AHS use *overlay model* of user knowledge (known also as overlay student model). The overlay model was also inherited from the field of ITS (Wenger, 1987). The key principle of the overlay model is that for each domain model concept, individual user knowledge model stores some data that is an estimation of the user knowledge level on this concept. In the simplest (and oldest) form it is a binary value (known – not known) that enables the model to represent user's knowledge as an overlay of domain knowledge. While some successful educational AHS (De Bra, 1996) use this classic form of overlay model, the majority of systems uses a weighted overlay model that can distinguish several levels of user's knowledge of a concept using a qualitative value (Brusilovsky, et al., 1998a; Grigoriadou, et al., 2001) (for example, good-average-poor), an integer quantitative value (for example, from 0 to 100) like shown on Figure 3 (Brusilovsky, et al., 1998b; De Bra, et al., 2001), or a probability that the user knows the concept (Henze, & Nejdl, 1999b; Specht, & Klemke, 2001).

A weighted overlay model of user knowledge can be represented as a set of pairs "concept–value", one pair for each domain concept. The overlay model is powerful and flexible because it can measure independently the user's knowledge of different concepts. Note that some versatile adaptive hypermedia systems that have integrated ITS components use more elaborate kind of overlay models to store multiple evidences about user level of knowledge separately (Brusilovsky, 1994; Weber, et al., 2001a).

In addition to concept-level overlay model, a number of AHS keep a page-level model also known as historic model. This model keeps some information about user visits to individual pages such as the number of visits or time spent on a page. While in some early adaptive hypermedia systems such as Manuel Excel (de La Passardiere, et al., 1992) the historic model was the only kind of student model used, modern AHS tend to ignore it or use it as a secondary source of adaptation. Still, ALE (Specht, Kravcik, Klemke, Pesin, & Hüttenhain, 2002), one of the most recent AH authoring systems has no concept overlay model and its adaptation bases solely on a historic model.

3.1.3. Modeling an educational goal

An ability to build a model of an educational goal is a very useful feature of an adaptive hypermedia system that extends its adaptivity and flexibility. This feature allows a teacher (or a student) to assign different educational goals to different students working with the same system. Using these individual goals, adaptive system can help different students to achieve different educational goals within the same course or set of knowledge. Currently the majority of educational AHS do not have a capability to model individual educational goals for students. In these systems all students have the same implicit educational goal - to learn all domain model concepts. In a few systems that support multiple educational goals (Brusilovsky, et al., 1998b; Grigoriadou, et al., 2001; Henze, et al., 1999a) the domain model offers a natural framework for goal modeling. This domain model-based approach to representing educational goals was also inherited by educational AHS from earlier works in the field of ITS (Brusilovsky, 1992; Elsom-Cook, & O'Malley, 1990; Vassileva, 1990). With this approach, an elementary educational goal is simply a target subset of domain concepts to be learned. This goal can be assigned to a student individually by the course author (Vassileva, & Deters, 1998), the teacher (Brusilovsky, et al., 1998b) or self-selected by the student (Brusilovsky, et al., 1998b; Grigoriadou, et al., 2001; Henze, et al., 1999a). In either case, with the goal mechanism in place, different students can pursue different educational goal within the same system. A natural way for the student to select an educational goal is to select a project, exercise, or other educational activity as a long-term goal (Henze, et al., 1999a) Then the prerequisite concepts of this activity form an individual educational goal. A more advanced elementary educational goal can be represented as a set of pairs: a concept and its target knowledge level. Elementary educational goals can be organized into more complex goals by being structured into sequences (goals have to be achieved one by one) or in trees (goals have to be achieved from leaves to the root). For example, in InterBook an individual learning goal was originally modeled as a sequence of sets (Figure 4) and later as stack of sets to enable a student to push a self-selected educational goal on the top of the stack (Schwarz, 1998). INSPIRE lets a student to pick up a structured educational goal for learning. This structured goal can be composed of a sequence of elementary sub-goals called layers each one composed as a set of concepts to be learned.

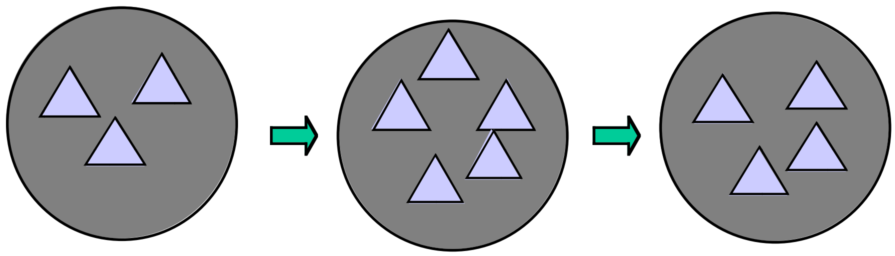


Figure 4. An individual educational goal can be modeled as a sequence of subsets of domain model concepts (shown as triangles).

3.2. Connecting knowledge with educational material

In Table 1, designing connections between the knowledge space and the hyperspace of educational material is listed as the last step in designing an educational AHS. We, however, review it before reviewing the hyperspace-structuring step since the approach to connecting “concepts and pages” significantly influence the hyperspace structure itself. The process of connecting domain knowledge with educational material is also known as indexing because specifying a set of underlying concepts for every page of educational material is very similar to indexing a page of content with a set of keywords. There are four aspects that are important to distinguish different indexing approaches: cardinality, granularity, navigation, and expressive power.

From the *cardinality* aspect, there are two essentially different cases: single concept indexing when each fragment of educational material is related to one and only one domain model concept and multi-concept indexing when each fragment can be related to many concepts. Single concept indexing is simpler and more intuitive for the authors. It has been applied in several AH authoring systems oriented to a non-expert user (Carmona, Bueno, EduardoGuzman, & Conejo, 2002; Vassileva, et al., 1998). Multi concept indexing is more powerful, but it makes the system more complex and requires more skilled authoring teams (Brusilovsky, et al., 1998b; De Bra, et al., 2001). It’s probably a good idea to choose single concept indexing whenever it is meaningful from educational point of view (i.e., smaller systems and simpler domains). At the same time, in many cases using a multi concept indexing is imposed by the nature of the domain. For example, in programming and mathematics, elementary constructs and operators are often selected as domain model concepts. In that case a hypermedia system that needs to have a reasonably precise indexing has to use a multi-concept indexing approach since most of the examples and problems involve several constructs and operators.

Expressive power concerns the amount of information that the authors can associate with every link between a concept and a page. Of course, the most important information is the very presence of the link. This case could be named as flat indexing and it is used in the majority of existing systems. Still some systems with large hyperspace and advanced adaptation techniques want to associate more information with every link by using roles and/or weights. Assigning a role to a link helps distinguish several kinds of connections between concepts and pages. For example, some systems want to distinguish a case when a page provides an introduction, a core explanation or a summary of a concept and a case when it provides a core explanation of it (Brusilovsky, 2000). Other systems use *prerequisite* role to mark the case when the concept is not presented on a page, but instead required for understanding it (Brusilovsky, et al., 1998b). Another way to increase expressive power of indexing is to specify weight of a link between a concept and a page. The weight may specify, for example, the percentage of knowledge about a concept presented on this page (De Bra, et al., 2001; Pilar da Silva, et al., 1998).

Granularity concerns the precision of indexing. The two most popular cases are indexing of the whole hypertext page with concepts and indexing of page fragments with concepts. There are also a few cases where the whole cluster of connected pages is indexed with the same set of concepts (Pérez, et al., 1995).

Navigation aspect is important to distinguish a case where a link between a concept and a page exists only on a conceptual level and is used only by internal adaptation mechanisms of the system from the case when each link also define a navigation path. My own position is that any conceptual link should be also used by navigation. In all systems developed by our group every concept has an external representation as a real or virtual page in the hyperspace and every indexing link between a page and a concept defines a navigation path. Thus, from any page with educational material students can navigate to all concepts connected with it and from each concept to all pages indexed with this concept. This approach creates very rich navigation opportunities and supports special style of navigation that I call concept-based navigation (Brusilovsky, & Schwarz, 1997).

Existing AH systems suggest various ways of indexing that differ in all aspects listed above. However, all this variety can be described in terms of three basic approaches that are described in the remaining part of this section. Systems using the same indexing approach have similar hyperspace structure and share specific adaptation techniques that are based on this structure. Thus the indexing approach selected by developers to a large extent defines the functionality of an AH system.

3.2.1. Concept-Based Hyperspace

The simplest approach to organizing connections between knowledge space and hyperspace is known as *concept-based hyperspace*. This approach is naturally appearing in any AH system that uses single-concept indexing. I distinguish simple and enhanced concept-based hyperspace. *Simple concept-based hyperspace* is used in systems that have exactly one page of educational material for every concept. With this approach, the hyperspace is built as an exact replica of the domain model (Figure 5). Each concept of the domain model is represented by exactly one node of the hyperspace, while the links between the concepts constitute main paths between hyperspace nodes (Brusilovsky, & Pesin, 1994; Brusilovsky, et al., 1993; Hohl, et al., 1996). A very attractive feature of this approach is that the educational hyperspace become perfectly structured just by indexing and does not really require any additional hypertext structuring approach analyzed in the next section. The simple concept based approach was quite popular among early educational AHS that have their roots in the ITS field. For these systems the concept-based hyperspace was simply the easiest and the most natural way to produce a well-structured hyperspace. Currently it is rarely used in educational AHS in its pure form because it requires each page of the hyperspace to be devoted to exactly one concept. It is very appropriate for developing encyclopedically structured learning material such as encyclopedias (Milosavljevic, 1997), manuals (Brusilovsky, et al., 1994), and glossaries (Brusilovsky, et al., 1998b), but too restrictive for practical Web-based education where multiple pages of educational material can be created to teach the same domain model concept. The most typical use of this approach in modern educational AHS is to represent a part of the overall hyperspace such as glossary of concepts in InterBook (Brusilovsky, et al., 1998b) or glossary of Lisp functions in ELM-ART (Weber, et al., 2001a).

The educational AHS with rich content and single-concept indexing can use an *enhanced concept-based hyperspace* design approach. With this design approach multiple pages describing the same concept are connected to this concept in both information space and hyperspace. Each concept has a corresponding "hub" page in the hyperspace. The concept hub page is connected by links to all educational hypertext pages related to this concept. The links can be typed and weighted (Grigoriadou, et al., 2001; Pilar da Silva, et al., 1998), though it is not necessary for using the approach. The student can navigate between hub concept pages along conceptual links and from hub pages to the pages with educational material. This approach could be used for creating relatively large AHS with quite straightforward structure and allows for a number of adaptation techniques (Grigoriadou, et al., 2001; Steinacker, et al., 1999).

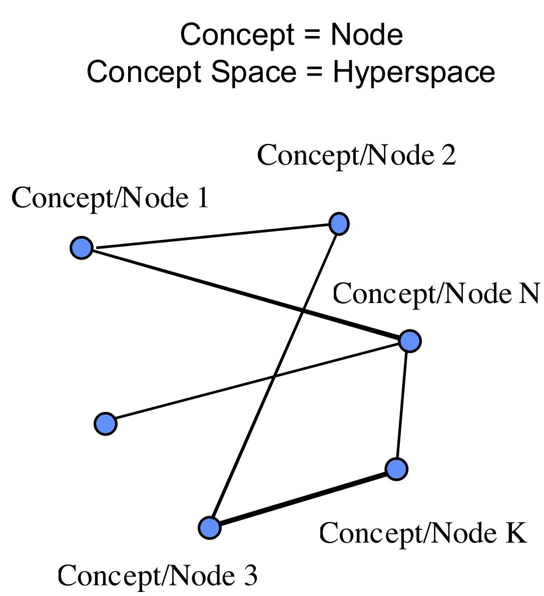


Figure 5. Simple concept-based hyperspace

The concept-based hyperspace design approach sets strong requirements to the domain model. It always requires a model with established links between concepts (preferably, several types of links) that will be used to establish hyperlinks.

Another restriction is that this approach can hardly be used "post-hoc" to turn an existing traditional hypermedia system into an AH system. It has to be used from the early steps of a hypermedia system design (Weber, et al., 2001b). However, this approach is very powerful and provides excellent opportunities for adaptation. With concept-based approach, the system knows exactly the type and content of each page and the type of each link. This knowledge can be used by various adaptive navigation support techniques. Annotation is the most popular technology here. For example, ISIS-Tutor (Brusilovsky, et al., 1994), ELM-ART (Weber, et al., 2001a), InterBook (Brusilovsky, et al., 1998b), INSPIRE (Grigoriadou, et al., 2001) use different kinds of link annotation to show the current educational state of the concept (not known, known, well known). ISIS-Tutor, ELM-ART, and a number of other systems use annotation to show that a concept page is not ready to be learned (i.e., its prerequisite concepts are not learned yet). Systems that support individual educational goals can use annotation to mark links to concept pages that belong to the current goal (Brusilovsky, et al., 1994). Hiding technology can be used to hide links to concept pages with not yet learned prerequisites (Brusilovsky, et al., 1994; Forcheri, et al., 2001) and concept pages that do not belong to the current educational goal (Brusilovsky, et al., 1994; Grigoriadou, et al., 2001).

Note that the concept-based hyperspace is just one of the possible design approaches for AHS with single concept indexing. There are a few known systems, especially among early AHS (Pérez, et al., 1995) with single concept indexing but without concept-based navigation. The concept-based hyperspace in these systems is not formed since concepts have no external hyperspace representation and/or links between concepts and pages are purely conceptual and not used for hyperspace navigation. However, once discovered, the concept-based hyperspace approach is becoming more and more popular and I recommend it to all developers of AHS who plan to use single-concept indexing.

3.2.2. Page Indexing

The most popular design approach for multi-concept indexing is page indexing. With this approach, the whole hypermedia page (node) is indexed with domain model concepts. In other words, links ate created between a page and each concepts that is related to the content of the page (Figure 6). The simplest indexing approach is flat content-based indexing when a concept is included in a page index if some part of this page presents the piece of knowledge designated by the concept (Brusilovsky, et al., 1994; Henze, et al., 1999a; Laroussi, & Benahmed, 1998). A more general but less often used way to index the pages is to add the role for each concept in the page index (role-based indexing). The most popular role is “prerequisite”: a concept is included in a page index if a student has to know this concept to understand the content of the page (Brusilovsky, et al., 1998b; Brusilovsky, et al., 1996a; De Bra, et al., 1998). Other roles can be used to specify the kind of contribution that the page is provided to learning this concept (introduction, main presentation, example, etc). Weights also can be used in multi-concept page indexing to show how much the page contributes to learning the concept (De Bra, et al., 2001).

Page indexing is a relatively simple approach. It can be applied even with vector domain models with no links between concepts (De Bra, 1996; Laroussi, et al., 1998). At the same time, indexing is a very powerful mechanism, because it provides the system with knowledge about the content of its pages. It opens the way for several adaptation techniques. With content-based indexing, the system knows which concepts are presented on a page. It can be used by various direct guidance, annotation and hiding techniques. Index-based guidance was used in ISIS-Tutor, ELM-ART and InterBook to recommend the “next best” page. Annotation was used to show the educational state of the page. For example, ELM-ART and ISIS tutor introduced a traffic light metaphor for annotation to distinguish and annotate the following important cases: the page is not ready to be learned (contains concepts which have not yet learned prerequisite concepts, red traffic light), the page is recommended (ready to be learned, contains concepts which are a part of the current learning goal, green traffic light). Prerequisite links can also be used to support simple adaptive presentation. For example, when LISP-Critic (Fischer, Mastaglio, Reeves, & Rieman, 1990) presents a page which has unknown prerequisite concepts, it inserts explanations of all the unknown concepts before the main content of the page. In the same context, ELM-ART and InterBook insert a warning message/image before the start of the page.

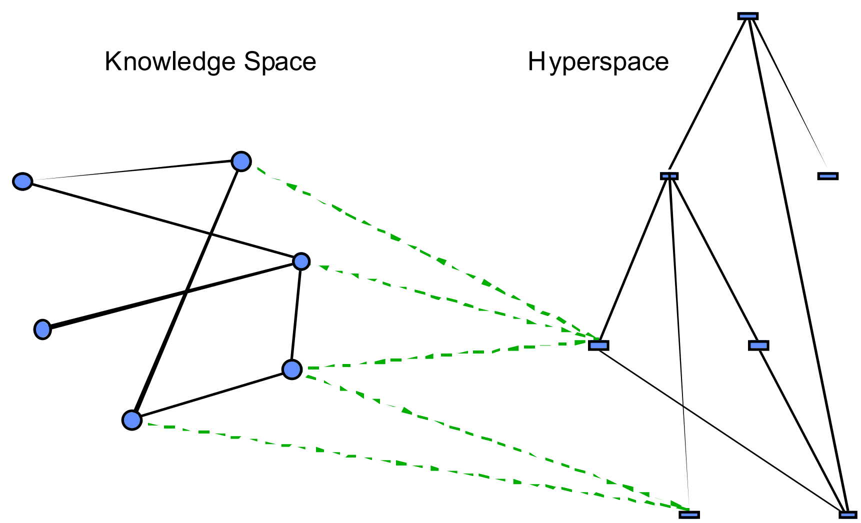


Figure 6. Multi-concept page indexing

3.2.3. Fragment Indexing

Fragment indexing is not a very popular indexing approach yet, but it is the most precise one. The idea of the approach is to divide the content of the hypermedia page into a set of fragments and to index some (or even all) of these fragments with domain model concepts which are related to the content of this fragment (Figure 7). Similarly to the page indexing approach it can be used even with a unstructured vector domain models. The difference is that indexing is done on a more fine-grained level. Generally, multi-concept indexing is used, though with smaller fragments it is often possible to use exactly one concept to index a fragment. In both cases the fragment indexing approach gives the system a more fine-grained knowledge about the content of the page: the system knows what is presented in each indexed fragment. This knowledge can be effectively used for advanced adaptive presentation. Depending on the level of user knowledge about the concepts presented in a particular fragment, the system can hide the fragment from the user (Boyle, & Encarnacion, 1994; De Bra, et al., 1998; Kobsa, Müller, & Nill, 1994; Stern, et al., 2000), shade it (Hothi, Hall, & Sly, 2000) or use an alternative way to present this fragment (Beaumont, 1994). Advanced adaptive presentation, of course, has its price – a more precise and more time-consuming indexing.

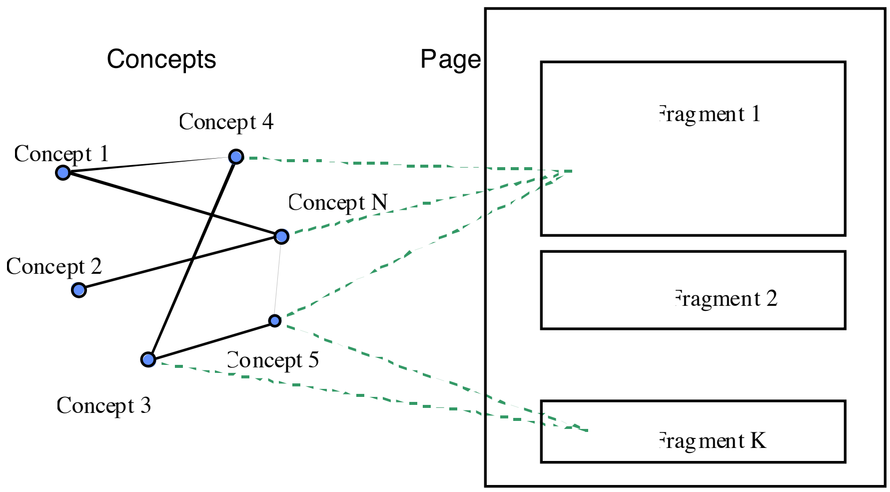


***Figure** 7. Fragment indexing approach*

A good example is provided by MetaDoc (Boyle, et al., 1994), the first system that implements this approach. MetaDoc not only indexes some text fragments with related concepts, but also distinguishes three types of fragments: general text, additional explanations, and low-level details. The system decides whether to present the text fragment to the user or to hide it depending on the user's level of knowledge of the indexing concepts. A user with good knowledge of a particular concept will always get additional explanations of this concept (which can be boring for that user) hidden and all low level details presented. On the contrary, the user with poor knowledge of a concept will always get additional explanations of this concept and all low-level details (too complicated for that user) will be hidden. The user with medium level knowledge will see both kinds of text fragments.

At the moment only three AH authoring systems - AHA (De Bra, et al., 1998; De Bra, et al., 2001), WHURLE (Brailsford, Stewart, Zakaria, & Moore, 2002), and ALE (Specht, et al., 2002) support an author in composing a page from indexed fragments. AHA and WHURLE support adaptive hiding of fragments and ALE also supports adaptive page structuring that is based on student learning styles.

3.2.4. Mixing the Approaches

The three above approaches are the basic ones for the organization of hyperspace in AH systems. These approaches, however, do not contradict each other. Moreover, they are really complimentary because they are based on the same domain and user models. Using more than one approach opens the way to use more adaptation techniques because each approach supports its own set of techniques. For example, ISIS-Tutor (Brusilovsky, et al., 1994) uses a knowledge-based approach to build a part of the hyperspace representing the concepts of a programming language. Another part of the hyperspace (where the pages are problems and examples) is organized by indexing pages with concepts. This organization lets ISIS-Tutor to use the adaptation techniques from two corresponding groups. InterBook (Brusilovsky, et al., 1998b) which is based on the same ideas as ISIS-Tutor also uses a combination of knowledge-based and page indexing approaches.

3.3. Structuring the Hyperspace

An important step in designing an educational AHS is structuring the hyperspace. The hyperspace is formed by content pages connected by navigation links. The topology of connection between pages is usually referred to as hyperspace structure. In modern educational AHS hyperspace structuring is a separate design step that is independent (in the sense of choosing the design approach) from the knowledge structuring step. This section reviews several known hyperspace structuring approaches.

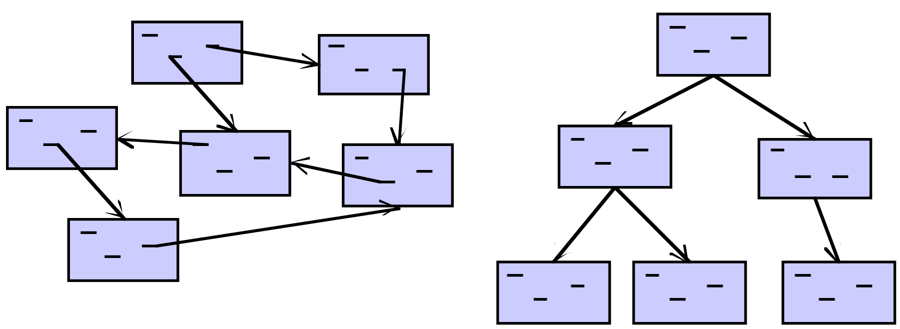


Figure 8. *Unstructured hyperspace (left) and hierarchically structured hyperspace (right)*

The first approach is *unstructured approach*, i.e. the case when this design step is skipped. It was the predominant "approach" in early educational AHS and it is still probably the most popular one. There are two reasons for that. First, concept-based hyperspace approach that is used in many education AHS produces quite rich and well-structured hyperspace that does not necessary require any other organization. Second, even in the systems that do not use concept-based approach the hyperspace is structured in some sense, since every page in it is connected to some other pages by classic hypertext reference links (Figure 8, left). There are educational AHS that use neither concept-based approach nor explicit hyperspace structure (De Bra, 1996; Pérez, et al., 1995) and even some design frameworks that impose no specific structuring requirements to the hyperspace and can work with an arbitrary structured hypertext (De Bra, et al., 1998). However, if no special strategy was used at the hyperspace design time, the resulting hyperspace may have a chaotic opportunistic structure that creates navigation and orientation problems to hypermedia users. It is not surprising that the vast majority of AH systems either use a concept-based approach or require the author to follow some explicit approach to structuring a hyperspace. Some recent systems oriented to Web-based education use both ways to obtain a rich well-structured hyperspace (Henze, et al., 1999a; Steinacker, et al., 1999).

The currently most popular approach to structuring the hyperspace is the hierarchical approach. ELM-ART (Brusilovsky, et al., 1996a) was the first adaptive hypermedia system that used a hierarchically structured hyperspace. This approach stimulated by the idea of an electronic book has been refined in InterBook (Brusilovsky, et al., 1998b) and later used in many educational AHS. With the hierarchical approach the hyperspace resembles a hierarchical structure of a textbook - with chapters, sections, and subsection (Figure 8). The hierarchy sets natural navigational paths from every node down to all descendant nodes and up to the ancestor node. Sequential "depth first" as well as "breadth first" navigational choices are usually also offered. One of the known benefits of a hierarchically structured hypertext is better navigation and orientation support in a hierarchy. This benefit of a hierarchical organization is especially important for categories of users that have navigation and orientation problems (Lin, 2003).

Another efficient approach to hyperspace structuring is "ASK" approach originally presented in (Bareiss, & Osgood, 1993). The goal of ASK approach is to build a "conversational" hyperspace with rhetorical links. With this approach a designer tries to anticipate a number of follow-up questions that the user may ask after visiting a page and provides links (each link is associated with a question) from the page to other pages in the hyperspace that may give answers to the questions. A generalization of this approach that uses a larger variety of rhetorical relationships is applied in MetaLinks (Murray, 2002) and Multibook (Steinacker, et al., 1999). We think that this promising approach will become more popular in the coming future.

Finally an interesting object-oriented approach to hyperspace structuring is used in KBS-Hyperbook system (Henze, et al., 1999a). The idea here is to build every page in the hyperspace as an external representation of some object. With this approach the hyperspace is turned into a network of objects that allows to use a standard object-oriented design approach and tools (such as UML) for hyperspace design. KBS-Hyperbook uses a good variety of objects of different level, such as chapter, lectures, examples, etc.

It is important to stress again that some of modern educational AHS often use a combination of several approaches to provide a better-structured hyperspace. In some cases, different approaches are used to structure different subspaces of the hyperspace. For example, in InterBook the set of glossary pages is structured as a concept-based hyperspace, while the remaining part of the hyperspace is formed by several hierarchically organized textbooks. In other cases two approaches can be used to "double-link" the same set of pages. For example, MetaLinks (Murray, 2002) uses a combination of the hierarchical and ASK approaches on the same set of textbook pages. Essentially, each of the structuring approaches allows the hyperspace designer to define a coherent set of links between pages. Thus the use of two approaches in parallel provides an opportunity to develop a more tightly interconnected hyperspace.

A special place should be devoted to the AHA! (De Bra, et al., 1998; De Bra, et al., 2001) structuring approach. The goal of this architecture is to model the information structure and functionality provided by a variety of adaptive hypermedia systems. To maintain the necessary level of generality, most recent versions of AHA completely blur the difference between pages and concepts. In AHA! both pages and concepts are modeled by AHA! pages/concepts (AHA! simply names them concepts). It allows the same mechanisms to be used for creating both content pages and domain concepts and the same linking and weight propagation mechanism to be used for defining the domain model, the hyperspace, and for linking these parts together. This generalized information model serves as a basis for a similarly generalized student model. Each page/concept may have any number of numeric properties (such as knowledge, interest, or simply number of visits) that can be propagated along different kinds of links (De Bra, Aerts, & Rousseau, 2002a; De Bra, Aerts, Smits, & Stash, 2002b). This generalized model can effectively simulate most known overlay and historic models while keeping enough potential for future development.

4. AUTHORING STAGE: HOW TO GET IT ALL INSIDE THE COMPUTER

Adaptive educational hypermedia is a reasonably mature field. Ten years of development brought to life dozens of experimental and practical systems and a good number of design approaches. Still, the field is probably too young to produce a good number of authoring tools, i.e., toolkits, systems, or shells that can be used by non-programming authors to develop an educational AHS. Two hands are still enough to count existing authoring systems (Table 2). The reason for that is reasonably clear. Before producing a real authoring tool, a research group has to develop an explicit design approach that usually requires developing one or more educational AHS. As shown on Table 2, existing authoring systems were produced by research groups that had some previous experience in developing AHS. In addition to that, there are several systems that were created as design frameworks such as CAMELEON (Laroussi, et al., 1998), PaKMaS (Süß, et al., 2000) RATH (Hockemeyer, et al., 1998), SKILL (Neumann, et al., 1998), ITMS (Mitsuhara, Kurose, Ochi, & Yano, 2001). These frameworks are application-independent and can be applied by their authors to rapid development of educational AHS in different domains. These design frameworks, however, have not yet reached the level of real tools that can be used by non-programming users outside of the original design team.

Generally, the basic steps of creating an educational AHS are not entirely different from steps in creating a regular hypertext system. In both cases the developer has to author the content objects and to specify the links between them. As in the case of generic hypertext and Web authoring, there are two major approaches used by authoring tools: the markup approach and GUI approach. In the markup approach the content of the pages and links between pages and concepts are authored in a regular word processor with the help of special markup language. With GUI (graphic user interface) approach they are created using special authoring user interface. In general, it could be a command-based, form-based or a direct manipulation interface, but all existing AHS authoring tools use form-based GUI.

The final product of both markup and GUI approaches is the same: the internal representation of knowledge and information in a some special form – a set of specially structured files or, more often, a database. This internal representation is used in the runtime by the AH engine to serve an AHS to the user. So, the authoring process is a bridge between the design and runtime stages, a process of getting the design inside the computer. The markup and GUI approaches provide two very different ways to achieve this goal. Since the difference between these two approaches is larger and more essential than the difference between ways to support the authoring steps listed in table 1, the following review of authoring tools is structured not by steps as the previous part, but by the authoring approaches.

Table 2. Authoring systems for developing adaptive educational hypermedia

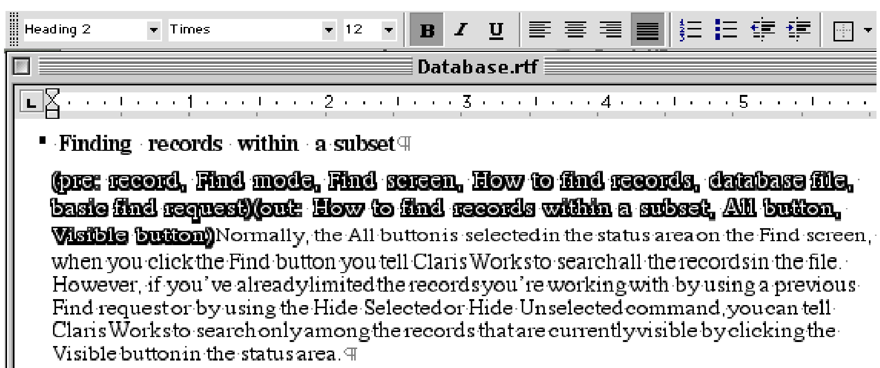
Authoring tool	*Earlier systems by the same authors*
InterBook (Brusilovsky, et al., 1998b)	ITEM/PG (Brusilovsky, et al., 1993), ISIS-Tutor (Brusilovsky, et al., 1994), ELM-ART (Brusilovsky, et al., 1996a)
Web DCG (Vassileva, et al., 1998) and DCG+GTE (Vassileva, 1998)	TOBIE (Vassileva, 1992), DCG (Diessel, Lehmann, & Vassileva, 1994)
AHA! (De Bra, et al., 2001) and AHA! 2.0 (De Bra, et al., 2002b)	2L670 (De Bra, 1996)
ACE (Specht, et al., 1998) and ALE (Specht, et al., 2002)	AST (Specht, Weber, Heitmeyer, & Schöch, 1997), ELM-ART II (Weber, & Specht, 1997), ADI (Schöch, Specht, & Weber, 1998)
NetCoach/ART-WEB (Weber, et al., 2001b)	ELM-ART (Brusilovsky, et al., 1996a), AST (Specht, et al., 1997), ELM-ART II (Weber, et al., 1997), ADI (Schöch, et al., 1998), ELM-ART III (Weber, et al., 2001a)
ECSAIWeb (Sanrach, et al., 2000)	ESCA (Grandbastien, & Gavignet, 1994)
MetaLinks (Murray, 2002)	Tectonica Interactive (Murray, Condit, & Haugsjaa, 1998)
SIGUE (Carmona, et al., 2002)	(Trella, Conejo, & Guzmán, 2000)

4.1. Markup-based authoring tools

In many application areas markup approach precedes the GUI approach. InterBook (Brusilovsky, Schwarz, & Weber, 1996b), the first known system for developing AHS used the markup approach. More exactly, InterBook applied a rather unusual kind of markup approach that was inherited from earlier ELM-ART system (Brusilovsky, et al., 1996a). An author can create an adaptive textbook completely in Microsoft Word™. Each textbook page or *section* in terms of InterBook was authored simply as a section in a Word document with a header on the top. All sections can reside in the same file or can be split in several Word files. The hierarchical structure of the hyperspace was defined as it is usually done in Word when creating hierarchical documents: by positioning descendant sections right after the ancestor section and assigning proper heading types to section headings.

Heading 1 was used for the top-level nodes in the tree, *Heading 2* for the second level nodes, and so on. The concept index for every section is authored in a special format with the use of a special font (Figure 9, top part). The index is divided into the prerequisite part (prefaced with *pre* tag on Figure 9) and the outcome part (prefaced *out* tag). Within each part, the concepts are simply listed by name. By using a few style conventions, we made Word work for the authors of an adaptive hyperbook as a kind of GUI. Different font types and styles work as markup operators. Since InterBook used a concept-based hyperspace approach, every concept may have an external representation as a glossary page. Glossary pages are created by the author as special subsections inside the last first-level section of the book named Glossary.

After the Word file with a textbook is created, it is saved in RTF format and converted to a real markup representation in the form of extended HTML (Figure 9, bottom part). What was expressed in Word with styles and fonts, is now expressed by markup commands. Since the markup commands had the form of specially structured HTML comments, the extended HTML file was still syntactically correct HTML and can be further edited/viewed using regular HTML tools. The extended HTML can be read directly by the InterBook runtime system that is based on Common LISP http server and converted to a LISP-based frame-styled internal representation.




```
<!--#2#-->
<h2><A NAME="Heading37"> Finding records within a
subset</A></h2>

<p><!--((pre: record, Find mode, Find screen, How to find
records, database file, basic find request)(out: How to find
records within a subset, All button, Visible button))-->
Normally, the All button is selected in the status area on the
Find screen, so when you click the Find button you tell
ClarisWorks to search all the records in the file. However, if
you've already limited the records you're working with by
using a previous Find request or by using the Hide Selected or
Hide Unselected command, you can tell ClarisWorks to search
only among the records that are currently visible by clicking
the Visible button in the status area.<p>
```

Figure 9. Authoring an adaptive hypertextbook in InterBook system. Top: Microsoft Word ™ view. Bottom: Extended HTML view. On both parts "pre" starts the list of prerequisite concepts and "out" – the list of outcome concepts.

Using unusual markup approaches was rather typical in the early "before XML" days of Web-based educational (WBE) systems. Other (non-adaptive) WBE authoring tools also applied various extensions of HTML to author structured educational hypertext and to extend it with metadata (Owen, & Makedon, 1997). Nowadays XML offers a standard way of markup-based authoring supported by various XML editors and parsers.

A good example of XML markup-based authoring is AHA! (De Bra, et al., 1998; De Bra, et al., 2001). AHA! uses a multi-concept weighted indexing approach. Every page may have several prerequisite and several outcome concepts. Link weights can specify the contribution this page can provide for learning of each outcome concepts or a degree of user interest in some concepts. Each page is created as an XML file that includes specially tagged index part and a page content. The page content is created in HTML format with XML extensions that are used to author conditional fragments (fragment indexing). In addition to that, a special XML file defines domain concepts and connections between them. AHA! doesn't use concept-based hyperspace approach. Concepts are nothing but names and there is no content associated with them. However, the concepts can be connected by non-typed weight-propagating relationships. These relationships can be used to represent several kinds of semantic relationships. For example, it is possible to model is-a relationship when demonstrated knowledge of lower concepts contribute to the knowledge of higher level concepts (e.g., knowledge about Belgian chocolate contributes to the knowledge about Belgian food). The domain model XML file lets the author to specify these relationships between concepts. The runtime engine of AHA! based on Java servlets reads the authored XML files, converts it into internal object representation and maintains the adaptive interaction with the user. As was mentioned above, AHA! Version 2.0 (De Bra, Brusilovsky, & Conejo, 2002c) extends this flexible approach even further by removing the difference between concepts and pages and allowing an arbitrary number of weight-propagating relationships between them. Yet, the XML-based authoring in the new AHA! is very similar to the older versions of AHA! (De Bra, et al., 1998; De Bra, et al., 2001). In fact, it is the use of the flexible XML approach that allows AHA! to evolve so rapidly.

The markup-based authoring approach is certainly very attractive because of the combination of low cost and expressive power. The presence of XML with a wide range of supporting tools makes is especially easy for the developers to get their design inside the computer. XML-based markup languages are used in several design frameworks such as PaKMaS (Süß, et al., 2000) with its Learning Material Markup Language (LMML), WHURLE (Brailsford, et al., 2002) with a different XML-based language, and a number of recent small AHS (Bonfigli, Casadei, & Salomoni, 2000; Forcheri, et al., 2001; Kurhila, & Sutinen, 2000). The problem with this approach is that it provides too much freedom and too little support for the authors. It does not really matter for domain-oriented systems made by professional members of the design teams, but it is a serious problem for teachers and domain experts who are the target users of authoring tools. Even with using syntax-driven XML editors or other approaches (like the one used in InterBook) when the author does not require to write all XML tags, there are too many chances to make hard-to-find errors. A good example is indexing pages with concepts that in any markup-based approach will imply writing a list of concept names somewhere inside the markup-based page description. If the user will make a small typo in the concept name, the system will consider it as a different concept and the indexing will be wrong. In InterBook it was the most typical error of non-programming authors. Thus, in addition to XML editors, a variety of checking tools has to be provided with a markup-based authoring system to open it to non-programming authors

4.2. Form-based authoring tools

GUI-based authoring tools are harder to develop, so they usually follow markup-based tools. As already mentioned, all existing GUI-based tools use form-based interface. This kind of interface provides very good support for the non-professionals. The most obvious way to use a form-based interface for authoring educational AHS is to provide one input form for entering information about a concept and another form for entering information about a page. Text boxes on this form will provide a simple way to enter text content and pop-up menus with concept or page names supporting authoring links between pages and concepts. In some cases only one form may be required. For example, MetaLinks (Murray, 2002) uses the concept-based hyperspace approach where every page represents a concept and use only one form. In other cases a separate form may be required for authoring different types of educational material. For example, KnowledgeTree authoring system (Brusilovsky, & Nijhawan, 2002) we use separate form for questions and examples that are special kinds of pages with learning material. ALE (Specht, et al., 2002) uses different forms to author learning material of different media types. The form based interface explicitly prompt the user what should be entered in each textbox and solves the problem of invalid input (i.e., typo in the concept name) by using menus and other form elements.

Figure 10. A form-based authoring interface in SIGUE system

A simple and straightforward example of practical form-based authoring is provided by SIGUE (Carmona, et al., 2002). SIGUE is an open corpus AHS - it relies exclusively on external Web-based educational content. SUGUE allows an author to organize and adaptively serve already existing content along the conceptual structure of the course. The core of a SIGUE course is a tree of concepts. The concepts are connected by prerequisite links that allow SIGUE to determine when each concept is ready to be learned and adaptively annotate links to the concept in ELM-ART/InterBook style. SIGUE uses a single-concept indexing paradigm: each external page is simply attached to one of the domain concepts. This straightforward organization requires a relatively simple authoring interface (Figure 10). Each domain concept can be authored in just three simple steps: (a) creating a new concept in a proper place within the concept three; (b) defining prerequisite links; (c) attaching external educational content.

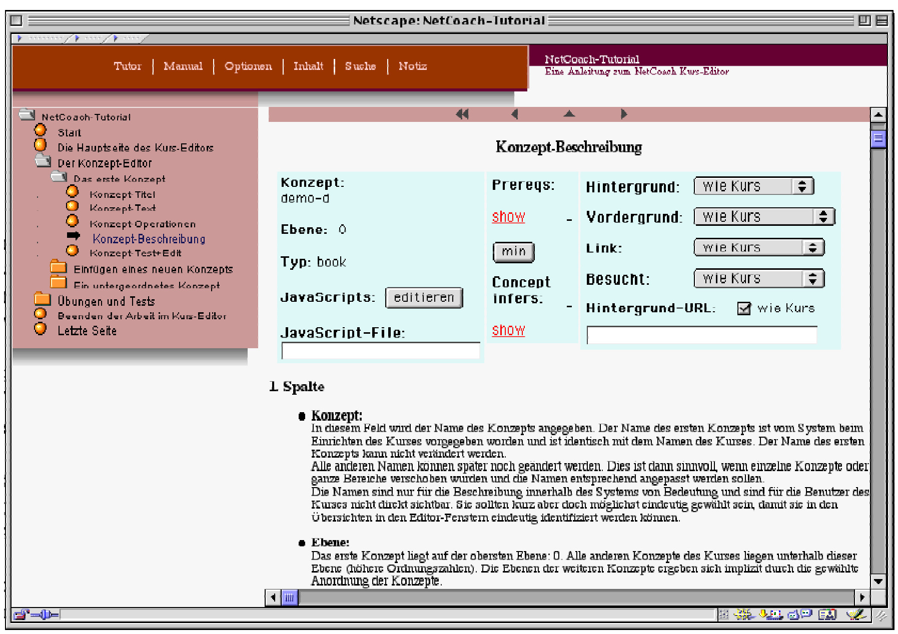


Figure 11. A form-based page/concept authoring interface in NetCoach system

A more complex example of a form-based interface is provided by NetCoach (Weber, et al., 2001b), the first commercially available tool for developing adaptive educational Web-based systems. NetCoach uses a concept-based hyperspace approach and supports both concept design and authoring and page authoring. As a results, its authoring interface is more complicated than in SIGUE. Pages/concepts in NetCoach also form a hierarchy, however links between concepts are more advanced than in SIGUE and include both prerequisite and inference (knowledge propagation) links. The form shown on (Figure 11) allows the author to provide page content (in HTML) as well as some presentation parameters, such as page background (*Hintergrund* in German) and behavior (in JavaScript). Other forms connected by links to the page/concept authoring form allow the author to specify prerequisite and inference links between the given page and other concept pages. The author can select an existing page to edit from an expandable menu (left side) or create a new descendant page of the current page. There are also separate forms for creating questions.

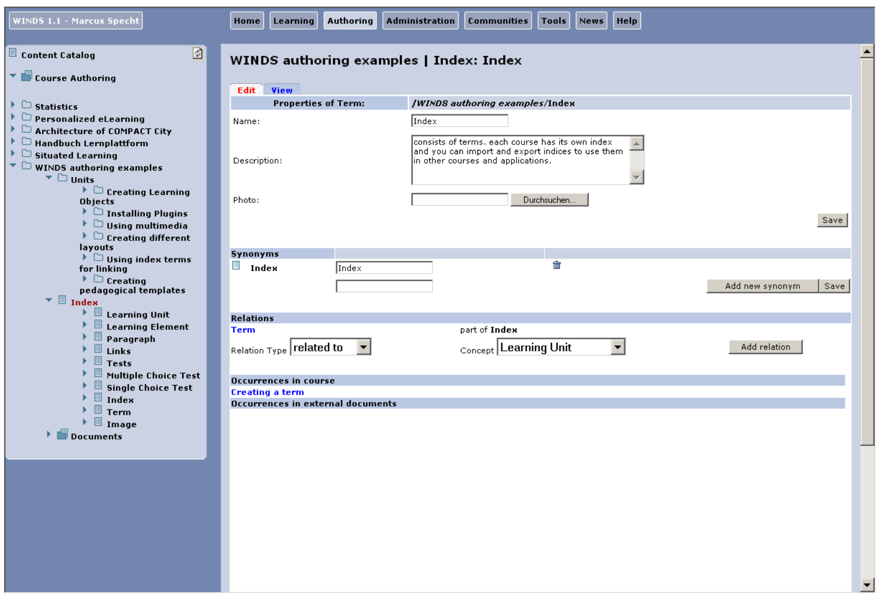


Figure12. Form-based term authoring in ALE

The currently most advanced form-based interface is offered by ALE (Specht, et al., 2002). ALE strives to provide a complete courseware management system for practical courses. To compete with other state-of-the art systems, it provides an extensive content authoring interface that supports object-reuse and various multimedia formats. This is a rather unusual for AH authoring systems that typically provide a limited to none support to content authoring. From AH point of view, however, ALE is straightforward since it supports a relatively low level of adaptivity. Similarly to InterBook, an information space in ALE is formed by a network of index terms (representing knowledge about the domain) and a tree of content pages. Pages can be composed from reusable fragments. They are indexed by terms using simple multi-concept approach. A form for term authoring (Figure 12) allows specifying links, a brief description, and synonyms. A group of page-authoring forms allows inserting a new page in any place within a hierarchy and composing it from multiple fragments of different media types. An interesting feature of ALE is automatic page indexing. Since every concept in ALE is simply a set of synonym terms, their presence in content pages can be discovered by a simple text analysis. Automatic page indexing is avoided in modern AHS due to its low precision. However, as we have mentioned above, ALE does not maintain concept-level user model and uses very simple adaptation mechanisms. In this context the possibly low precision of indexing does not influence the adaptation.

As demonstrated by the above examples, nearly every AHS feature can be authored using a form-based interface, but it comes for a price. What requires just another tag in markup-based tools may require several interface widgets and overhead of positioning all these widgets in the main form and its sub-forms. While simple solutions can be authored with a relatively simple interface, an interface for the systems with rich and competitively looking content or rich information structure tend to be complicated and expensive to develop.

To avoid problems with developing a more advanced content, early AH authoring systems attempted to rely on existing external tools. For example, InterBook authors were expected to use Microsoft Word formatting capability. MetaLinks (Murray, 2002) has used creatively a form-based authoring interface of Macintosh *Filemaker Pro* database. Most recent authoring systems such SIGUE tend to ignore the issue of content development completely. Indeed, it's just makes no sense to compete with modern Web content authoring tools such as *Macromedia Dreamweaver* or *Adobe GoLive*. Even ALE with its most advanced content authoring interface supports the use of externally authored content elements.

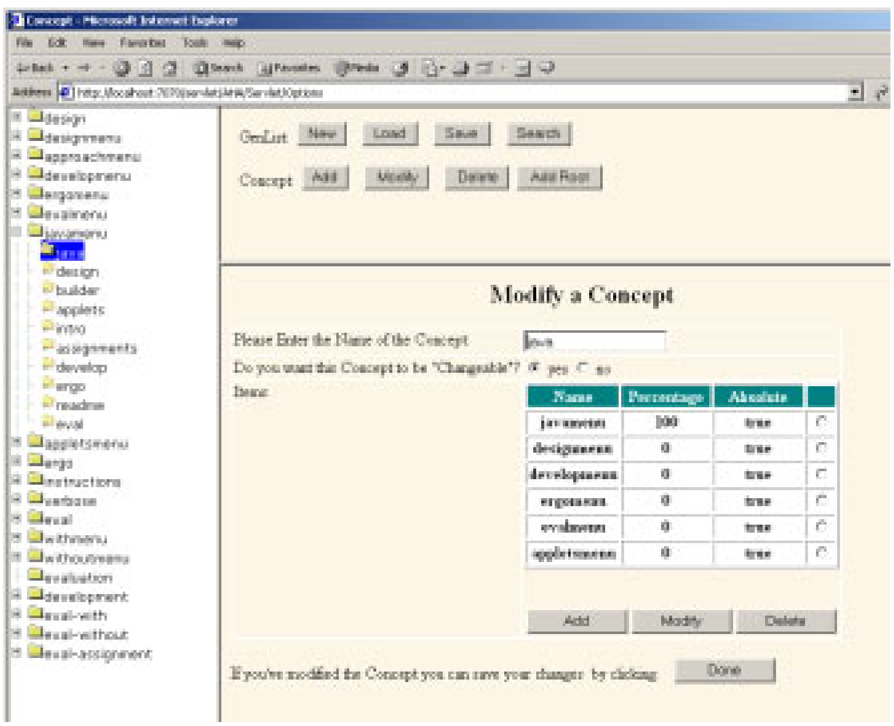


Figure 13. Domain model authoring in AHA.

Unfortunately, there are no form-based external tools that can help adaptive hypermedia authors to develop rich information structures. Here AH authoring tools have to provide full support. While simple structures can be successfully supported by a form-based interface as demonstrated by the systems above, more advanced structures still can't avoid a markup-based approach. In this context, a combination of a markup-based interface and a form-based interface can provide a good compromise between development cost and level of author's support. Some components of the target system where support and error prevention is essential can be created with a form-based interface, while the rest is authored with a markup approach. This combined interface will suite well an intermediate-level author: not completely computer-naive, but not one of the original tool developers either. AHA! (De Bra, et al., 1998) provides a good example. While the full power of AHA! (especially Version 2.0) is only available through the XML-based markup language, the system now also provide some form-based authoring tools (Figure 13) for the development of domain model (De Bra, et al., 2001). The authoring interface shown in Figure 13 allows an author to define new concepts, links between them, and weight propagation along the links. As it should be in the combined interface, the form-based editor produces a domain model description in the system's markup language that can be further extended by more demanding users.

5. A VIEW TO THE FUTURE

The previous sections of the review presented the current state of AH authoring. What can we expect in the next five years? I think that we may expect a splash of work in the area. The mature state of adaptive hypermedia as a field with a good number of established techniques and the need for more adaptive systems for Web-based education clearly serve as two major driving forces. These forces have brought many new teams in the area of adaptive Web-based educational hypermedia. A number of new frameworks have been introduced just within the last three years. It's natural to expect that several frameworks will be elaborated to the level of end-user oriented authoring tools just in a few years. The experience of the few existing authoring tools will help to shape out new tools and new generations of old tools. We may expect better markup-based and form-based tools.

I also expect the appearance of a new kind of "really graphic" user interface design tools. It's a definite contradiction that there is no graphic interface for getting highly graphical results of AHS design into the computer. The first graphic authoring tool that we will see really soon is a graphic editor of concept networks. With this editor the designer will be able to design and author connections between concepts just by placing concepts on a working window and connecting with a drag-and-drop interface. This tool will support both the design and authoring stages. A number of similar network editors exist currently in the field of concept-mapping and business presentation design, but all these tools are useful only with relatively small maps (under 50 nodes) and the main result of the work is the visual presentation of the network. As a result, these tools now can only be used during the design stage and for developing small networks. Future concept network editing tools will be very similar, but their main product will be an internal representation of the network in a database or XML markup form[2]. Gradually, they will also handle networks of 100 and more concepts. After concept editors we may expect the appearance of similar graphic tools for structuring the hyperspace and for indexing pages or fragments with domain model concepts.

---

[2] This prediction turned reality sooner than the author expected. While this book was in the process of publication, the first concept-network authoring tool for adaptive hypermedia system has been reported in (De Bra, et al., 2002a).

I also expect a gradual merge between two close approaches to authoring educational material. The one (reviewed in this paper) concerns authoring for adaptive Web-based educational systems. The other is the "new wave" in Web-based courseware authoring centered on learning objects, courseware reuse, metadata and standards. The authoring paradigm of this approach is quite similar to the one of adaptive systems. It is based on creating pools of learning objects of different level and indexing it with metadata. A Web/hypertext page with educational material is a typical learning object. One of the important kinds of metadata are learning objectives that are very similar to domain concepts from indexing point of view (they also can be used as outcomes or prerequisites of an educational object). Thus concept-indexed pages with educational material are not different from other reusable content objects indexed with metadata. With several adopted or coming standards in the area of metadata and metadata-enabled material (AICC, IEEE LTSC, IMS, SCORM) the metadata-based courseware re-use is reaching its maturity. More and more standard-compliant authoring tools are becoming available. We may expect that a number of authoring tools created for metadata-based courseware reuse approach will be quite useful for the authors of educational AHS. For example, there are some graphic tools for topic map editing in the courseware reuse field that are almost ready to be used as concept network editors (Coffey, & Cañas, 2001; Steinacker, et al., 2001). There are also some very good indexing tools that are almost ready to be used as concept indexers (Verhoeven, Cardinaels, Van Durm, Duval, & Olivié, 2001).

From the authoring and storage side adaptive hypermedia and courseware reuse approaches are quite similar. The difference is how the indexed material is used. In the courseware reuse approach the metadata are used by a search engine that helps a teacher to find and re-use educational material for his or her manually created course. An adaptive educational system uses it to decide whether a page of learning material is relevant in the current context and how to present the page and the links to it. There are no contractions between these goals so approaches may eventually use the same indexed educational material. There are already a few attempts to combine the adaptive hypermedia and metadata-based courseware re-use approaches in one framework (Brailsford, et al., 2002; Fischer, & Steinmetz, 2000; Specht, et al., 2001; Specht, et al., 2002; Steinacker, et al., 2001; Süß, et al., 2000). In the near future we should expect more powerful authoring systems and frameworks that combine adaptive hypermedia and courseware reusability ideas.

6. A VIEW TO THE PAST

This review is based on my personal (and probably subjective) view on AHS design based on my own work on adaptive hypermedia for about 10 years. Over these years my view on adaptive hypermedia and my understanding of it has broadened quite significantly. I was fortunate to work on adaptive hypermedia problems with several teams that were driven by different goals. I think, it has helped me to consider adaptive educational hypermedia from several prospects and to avoid a narrow and pragmatic view to it. To make my current prospect and viewpoint more clear to a reader, I risk to provide a brief history of my research on adaptive hypermedia. I hope, this section will explain how the change of projects and prospects has helped me to form the generalized view on adaptive hypermedia presented in this paper..

My research on adaptive educational hypermedia has started in 1991 at the Moscow State University with a group of students as an extension of my earlier work on curriculum sequencing in ITS (Brusilovsky, 1992). Our main focus was *adaptive navigation support,* currently one of the major AH technologies. We have developed two educational AH systems ITEM/PG (Brusilovsky, et al., 1993) and ISIS-Tutor (Brusilovsky, et al., 1994) and explored several ways of adaptive navigation support: direct guidance in the form of "teach me" button, adaptive annotation, and adaptive link removal. Our vision was that adaptive navigation support in educational hypermedia is "best of both worlds". Choosing the next task in an ITS with sequencing is based on machine intelligence. Choosing the next task in traditional hypermedia is based on human intelligence. Adaptive navigation support is an interface that can integrate the power of machine and human intelligence: a user is free to make a choice while still seeing an opinion of an intelligent system. Following this vision we considered adaptive hypermedia as a necessary *component of every ITS* (Brusilovsky, et al., 1993) and developed a specific authoring approach and a toolkit for developing adaptive hypermedia components for ITS (Brusilovsky, 1997).

The work on ELM-ART (Brusilovsky, et al., 1996a; Weber, et al., 2001a), one of the first practical Web-based ITS helped me to extend and re-consider the original approach as well as to adopt it to the hypermedia-centered Web context. Our team has adopted a new *electronic textbook* vision for an adaptive educational hypermedia systems. This approach and this vision have been refined and extended during our subsequent work on InterBook system (Brusilovsky, et al., 1998b), the first authoring system for adaptive educational hypermedia.

Two years of practical work in Carnegie Technology Education, a company focused on Web-based education and several years of participation in IEEE Learning Technology Standardization committee helped me to understand and appreciate the problems of WBE from both academic and industrial prospects. While my understanding of “what can be adapted” in adaptive educational hypermedia has not been significantly changed, my understanding of what is an adaptive educational hypermedia system and how it can be designed was considerably broadened and affected by the needs of practical Web-based education. I now consider adaptive hypermedia as a generic way to organize and design any hypermedia-based educational resource. It includes our earlier paradigms such as hypermedia-based ITS and adaptive electronic textbooks as well as a number of other known paradigms such as adaptive Web-based courses, pools of re-usable educational objects and even distributed educational hypermedia. From my point of view, all known types of adaptive hypermedia systems, while looking different to a casual observer, are really very similar. They can be viewed as instantiations of the same “universal” design approach and use subsets of the same collection of AH techniques. This is the viewpoint that I have attempted to communicate in this paper.

7. CONCLUSIONS

The availability of authoring frameworks and tools is gradually changing the situation in the field of adaptive hypermedia. Just a few years ago designing an adaptive hypermedia system was a large time-consuming endeavor that could only be attempted by skilled teams with programming expertise. With design frameworks and tools at hand, a less prepared teams and even a single teacher can create a meaningful adaptive hypermedia system. The challenge for the authors is to be aware about the variety of available tools and pick the right tool for the work. We hope that the anatomy of educational AHS design and a survey of existing approaches and tools provided in this paper will be helpful for the researchers in the field of adaptive hypermedia and authors of practical educational AHS.

8. REFERENCES

Bareiss, R., & Osgood, R. (1993). Applying AI models to the design of exploratory hypermedia systems. *Proceedings of Fifth ACM Conference on Hypertext,* 1993. Seattle, WA, ACM. - pp. 94-105.

Beaumont, I. (1994). User modeling in the interactive anatomy tutoring system ANATOM-TUTOR. *User Modeling and User-Adapted Interaction*, **4** (1), 21-45.

Bonfigli, M. E., Casadei, G., & Salomoni, P. (2000). Adaptive Intelligent Hypermedia using XML. *Proceedings of ACM Symposium on Applied Computing,* March 19-21, 2000. Como, Italy, http://www.acm.org/conferences/sac/sac00/Proceed/FinalPapers/WW-06.

Boyle, C., & Encarnacion, A. O. (1994). MetaDoc: an adaptive hypertext reading system. *User Modeling and User-Adapted Interaction*, **4** (1), 1-19.

Brailsford, T. J., Stewart, C. D., Zakaria, M. R., & Moore, A. (2002). Autonavigation, links, and narrative in an adaptive Web-based integrated learning environment. *Proceedings of World Wide Web 2002 Conference,* May 7-11, 2002. Honolulu, HI, - pp. .

Brusilovsky, P. (1994). Student model centered architecture for intelligent learning environment. *Proceedings of Fourth International Conference on User Modeling,* 15-19 August 1994. Hyannis, MA, MITRE. - pp. 31-36, available online at http://www2.sis.pitt.edu/~peterb/papers/UM94.html.

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

Brusilovsky, P. (1997). Integrating hypermedia and intelligent tutoring technologies: from systems to authoring tools. In P. Kommers, A. Dovgiallo, V. Petrushin, & P. Brusilovsky (Eds.), *New media and telematic technologies for education in Eastern European countries* (pp. 129-140). Enschede: Twente University Press.

Brusilovsky, P. (2000). Concept-based courseware engineering for large scale Web-based education. In G. Davies, & C. Owen (Eds.), *Proceedings of WebNet'2000, World Conference of the WWW and Internet,* Oct. 30 - Nov. 4, 2000. San Antonio, TX, AACE. - pp. 69-74.

Brusilovsky, P., & Anderson, J. (1998a). ACT-R electronic bookshelf: An adaptive system for learning cognitive psychology on the Web. In H. Maurer, & R. G. Olson (Eds.), *Proceedings of WebNet'98, World Conference of the WWW, Internet, and Intranet,* November 7-12, 1998. Orlando, FL, AACE. - pp. 92-97.

Brusilovsky, P., Eklund, J., & Schwarz, E. (1998b). Web-based education for all: A tool for developing adaptive courseware. *Computer Networks and ISDN Systems*, **30** (1-7), 291-300.

Brusilovsky, P., & Nijhawan, H. (2002). A Framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities. In M. Driscoll, & T. C. Reeves (Eds.), *Proceedings of World Conference on E-Learning, E-Learn 2002,* October 15-19, 2002. Montreal, Canada, AACE. - pp. 154-161.

Brusilovsky, P., & Pesin, L. (1994). An intelligent learning environment for CDS/ISIS users. In J. J. Levonen, & M. T. Tukianinen (Eds.), *Proceedings of The interdisciplinary workshop on complex learning in computer environments (CLCE94),* May 16-19, 1994. Joensuu, Finland, EIC. - pp. 29-33, available online at http://cs.joensuu.fi/~mtuki/www_clce.270296/Brusilov.html.

Brusilovsky, P., Pesin, L., & Zyryanov, M. (1993). Towards an adaptive hypermedia component for an intelligent learning environment. In L. J. Bass, J. Gornostaev, & C.

Unger (Ed.), *3rd International Conference on Human-Computer Interaction, EWHCI'93* (Vol. 753, pp. 348-358). Berlin: Springer-Verlag.
Brusilovsky, P., & Schwarz, E. (1997). Concept-based navigation in educational hypermedia and its implementation on WWW. In T. Müldner, & T. C. Reeves (Eds.), Educational Multimedia/Hypermedia and Telecommunications, 1997, *Proceedings of ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications,* June 14-19, 1997. Calgary, Canada, AACE. - pp. 112-117.
Brusilovsky, P., Schwarz, E., & Weber, G. (1996a). ELM-ART: An intelligent tutoring system on World Wide Web. In C. Frasson, G. Gauthier, & A. Lesgold (Ed.), *Third International Conference on Intelligent Tutoring Systems, ITS-96* (pp. 261-269). Berlin: Springer Verlag, available online at http://www.contrib.andrew.cmu.edu/~plb/ITS96.html.
Brusilovsky, P., Schwarz, E., & Weber, G. (1996b). A tool for developing adaptive electronic textbooks on WWW. In H. Maurer (Eds.), *Proceedings of WebNet'96, World Conference of the Web Society,* October 15-19, 1996. San Francisco, CA, AACE. - pp. 64-69, available online at http://www.contrib.andrew.cmu.edu/~plb/WebNet96.html.
Brusilovsky, P. L. (1992). A framework for intelligent knowledge sequencing and task sequencing. In C. Frasson, G. Gauthier, & G. I. McCalla (Ed.), *Second International Conference on Intelligent Tutoring Systems, ITS'92* (pp. 499-506). Berlin: Springer-Verlag.
Brusilovsky, P. L. (1993). Towards an intelligent environment for learning introductory programming. In G. Dettori, B. du Boulay, & E. Lemut (Eds.), *Cognitive Models and Intelligent Environments for Learning Programming* (pp. 114-124). Berlin: Springer-Verlag.
Carmona, C., Bueno, D., EduardoGuzman, & Conejo, R. (2002). SIGUE: Making Web Courses Adaptive. In P. De Bra, P. Brusilovsky, & R. Conejo (Eds.), *Proceedings of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002),* May 29-31, 2002. Málaga, Spain, - pp. 376-379.
Coffey, J. W., & Cañas, A. J. (2001). Tools to Foster Course and Content Reuse in Online Instructional Systems. In W. Fowler, & J. Hasebrook (Eds.), *Proceedings of WebNet'2001, World Conference of the WWW and Internet,* October 23-27, 2001. Orlando, FL, AACE. - pp. 207-213.
De Bra, P., Aerts, A., & Rousseau, B. (2002a). Concept Relationship Types for AHA! 2.0. In M. Driscoll, & T. C. Reeves (Eds.), *Proceedings of World Conference on E-Learning, E-Learn 2002,* October 15-19, 2002. Montreal, Canada, AACE. - pp. 1386-1389.
De Bra, P., Aerts, A., Smits, D., & Stash, N. (2002b). AHA! Version 2.0: More Adaptation Flexibility for Authors. In M. Driscoll, & T. C. Reeves (Eds.), *Proceedings of World Conference on E-Learning, E-Learn 2002,* October 15-19, 2002. Montreal, Canada, AACE. - pp. 240-246.
De Bra, P., Brusilovsky, P., & Conejo, R. (Ed.). (2002c). *Adaptive Hypermedia and Adaptive Web-based Systems, AH2002*. Berlin: Springer-Verlag.
De Bra, P., & Calvi, L. (1998). AHA! An open Adaptive Hypermedia Architecture. *The New Review of Hypermedia and Multimedia*, **4** 115-139.
De Bra, P., & Ruiter, J.-P. (2001). AHA! Adaptive hypermedia for all. In W. Fowler, & J. Hasebrook (Eds.), *Proceedings of WebNet'2001, World Conference of the WWW and Internet,* October 23-27, 2001. Orlando, FL, AACE. - pp. 262-268.
De Bra, P. M. E. (1996). Teaching Hypertext and Hypermedia through the Web. *Journal of Universal Computer Science*, **2** (12), 797-804, available online at http://www.iicm.edu/jucs_2_12/teaching_hypertext_and_hypermedia.
de La Passardiere, B., & Dufresne, A. (1992). Adaptive navigational tools for educational hypermedia. In I. Tomek (Ed.), *ICCAL'92, 4-th International Conference on Computers and Learning* (pp. 555-567). Berlin: Springer-Verlag.
Diessel, T., Lehmann, A., & Vassileva, J. (1994). Individualised course generation: A marriage between CAL and ICAL. *Computers and Education*, **22** (1/2), 57-64.

Elsom-Cook, M. T., & O'Malley, C. (1990). ECAL: Bridging the gap between CAL and intelligent tutoring systems. *Computers and Education*, **15** (1), 69-81.

Fischer, G., Mastaglio, T., Reeves, B., & Rieman, J. (1990). Minimalist explanations in knowledge-based systems. *Proceedings of 23-th Annual Hawaii International Conference on System Sciences,* January 2-5, 1990. Kailua-Kona, HI, IEEE. - pp. 309-317.

Fischer, S., & Steinmetz, R. (2000). Automatic creation of exercises in adaptive hypermedia learning systems. *Proceedings of Eleventh ACM Conference on Hypertext and hypermedia (Hypertext 2000),* May 30 - June 3, 2000. San Antonio, TX, ACM Press. - pp. 49-55.

Forcheri, P., Molfino, M. T., Moretti, S., & Quarati, A. (2001). An Approach to the Development of Re-Usable and Adaptive Web Based Courses. In W. Fowler, & J. Hasebrook (Eds.), *Proceedings of WebNet'2001, World Conference of the WWW and Internet,* October 23-27, 2001. Orlando, FL, AACE. - pp. 1043-1048.

Gonschorek, M., & Herzog, C. (1995). Using hypertext for an adaptive helpsystem in an intelligent tutoring system. In J. Greer (Eds.), 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. 274-281.

Grandbastien, M., & Gavignet, E. (1994). ESCA: An environment to design and instantiate learning material. In T. de Jong, & L. Sarti (Eds.), *Design and production of multimedia and simulation-based learning material* (pp. 31-44). Dordrecht: Kluwer Academic Publishers.

Grigoriadou, M., Papanikolaou, K., Kornilakis, H., & Magoulas, G. (2001). INSPIRE: An INtelligent System for Personalized Instruction in a Remote Environment. In P. D. Bra, P. Brusilovsky, & A. Kobsa (Eds.), *Proceedings of Third workshop on Adaptive Hypertext and Hypermedia,* July 14, 2001. Sonthofen, Germany, Technical University Eindhoven. - pp. 13-24.

Henze, N., Naceur, K., Nejdl, W., & Wolpers, M. (1999a). Adaptive hyperbooks for constructivist teaching. *Künstliche Intelligenz*, (4), 26-31.

Henze, N., & Nejdl, W. (1999b). *Student modeling for KBS Hyperbook system using Bayesian networks* (Report # . University of Hannover, available online at http://www.kbs.uni-hannover.de/paper/99/adaptivity.html.

Hockemeyer, C., Held, T., & Albert, D. (1998). RATH - A relational adaptive tutoring hypertext WWW-environment based on knowledge space theory. In C. Alvegård (Eds.), *Proceedings of CALISCE'98, 4th International conference on Computer Aided Learning and Instruction in Science and Engineering,* June 15-17, 1998. Göteborg, Sweden, - pp. 417-423.

Hohl, H., Böcker, H.-D., & Gunzenhäuser, R. (1996). Hypadapter: An adaptive hypertext system for exploratory learning and programming. *User Modeling and User-Adapted Interaction*, **6** (2-3), 131-156.

Hothi, J., Hall, W., & Sly, T. (2000). A study comparing the use of shaded text and adaptive navigation support in adaptive hypermedia. In P. Brusilovsky, O. Stock, & C. Strapparava (Ed.), *Adaptive Hypermedia and Adaptive Web-based systens* (pp. 335-342). Berlin: Springer-Verlag.

Kay, J., & Kummerfeld, R. J. (1994). An individualised course for the C programming language. *Proceedings of Second International WWW Conference,* 17-20 October, 1994. Chicago, IL, - pp. , available online at http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Educ/kummerfeld/kummerfeld.html.

Kobsa, A., Müller, D., & Nill, A. (1994). KN-AHS: An adaptive hypertext client of the user modeling system BGP-MS. *Proceedings of Fourth International Conference on User Modeling,* 15-19 August 1994. Hyannis, MA, MITRE. - pp. 31-36.

Kurhila, J., & Sutinen, E. (2000). From intelligent tutoring systems to intelligent learning materials. In J. Bordeau, & R. Heller (Eds.), Educational Multimedia, Hypermedia and Telecommunications, 2000, *Proceedings of ED-MEDIA'2000 - World Conference on*

Educational Multimedia, Hypermedia and Telecommunications, June 26 - July 1, 2000. Montréal, Canada, AACE. - pp. 511-516.

Laroussi, M., & Benahmed, M. (1998). Providing an adaptive learning through the Web case of CAMELEON: Computer Aided MEdium for LEarning on Networks. In C. Alvegård (Eds.), *Proceedings of CALISCE'98, 4th International conference on Computer Aided Learning and Instruction in Science and Engineering,* June 15-17, 1998. Göteborg, Sweden, - pp. 411-416.

Lin, D.-Y. M. (2003). Hypertext for the aged: effects of text topologies. *Computers in Human Behavior*, **19** (2003), 201-209.

Milosavljevic, M. (1997). Augmenting the user's knowledge via comparison. In A. Jameson, C. Paris, & C. Tasso (Ed.), *6th International Conference on User Modeling, UM97* (pp. 119-130). Wien: SpringerWienNewYork.

Mitsuhara, H., Kurose, Y., Ochi, Y., & Yano, Y. (2001). ITMS: Individualized Teaching Material System-adaptive integration of web pages distributed in some servers. *Proceedings of ED-MEDIA'2001 - World Conference on Educational Multimedia, Hypermedia and Telecommunications,* June 25-30, 2001. Tampere, Finland, AACE. - pp. 1333-1338.

Murray, T. (2001). Characteristics and Affordances of Adaptive Hyperbooks. In W. Fowler, & J. Hasebrook (Eds.), *Proceedings of WebNet'2001, World Conference of the WWW and Internet,* October 23-27, 2001. Orlando, FL, AACE. - pp. 899-904.

Murray, T. (2002). MetaLinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks. *International Journal of Artificial Intelligence in Education*, **13** (1), To appear.

Murray, T., Condit, C., & Haugsjaa, E. (1998). MetaLinks: A preliminary framework for concept-based adaptive hypermedia. *Proceedings of Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems (ITS'98),* August 16-19, 1998. San Antonio, TX, - pp. , available online at http://www-aml.cs.umass.edu/~stern/webits/itsworkshop/murray.html.

Neumann, G., & Zirvas, J. (1998). SKILL - A scallable internet-based teaching and learning system. In H. Maurer, & R. G. Olson (Eds.), *Proceedings of WebNet'98, World Conference of the WWW, Internet, and Intranet,* November 7-12, 1998. Orlando, FL, AACE. - pp. 688-693, available online at http://nestroy.wi-inf.uni-essen.de/Forschung/Publikationen/skill-webnet98.ps.

Oberlander, J., O'Donell, M., Mellish, C., & Knott, A. (1998). Conversation in the museum: experiments in dynamic hypermedia with the intelligent labeling explorer. *The New Review of Multimedia and Hypermedia*, **4** 11-32.

Owen, C. B., & Makedon, F. (1997). ASML: Web authoring by site, not by hidnsight. In T. Müldner, & T. C. Reeves (Eds.), Educational Multimedia/Hypermedia and Telecommunications, 1997, *Proceedings of ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications,* June 14-19, 1997. Calgary, Canada, AACE. - pp. 677-682.

Pérez, T., Gutiérrez, J., & Lopistéguy, P. (1995). An adaptive hypermedia system. In J. Greer (Eds.), 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. 351-358.

Pilar da Silva, D., Durm, R. V., Duval, E., & Olivié, H. (1998). *Concepts and documents for adaptive educational hypermedia: a model and a prototype*. Computing Science Reports, Eindhoven: Eindhoven University of Technology, 35-43.

Sanrach, C., & Grandbastien, M. (2000). ECSAIWeb: A Web-based authoring system to create adaptive learning systems. In P. Brusilovsky, O. Stock, & C. Strapparava (Ed.), *Adaptive Hypermedia and Adaptive Web-based Systems, AH2000* (pp. 214-226). Berlin: Springer-Verlag.

Schöch, V., Specht, M., & Weber, G. (1998). "ADI" - an empirical evaluation of a tutorial agent. In T. Ottmann, & I. Tomek (Eds.), Educational Multimedia/Hypermedia and

Telecommunications, 1998, *Proceedings of ED-MEDIA/ED-TELECOM'98 - 10th World Conference on Educational Multimedia and Hypermedia and World Conference on Educational Telecommunications,* June, 20-25, 1998. Freiburg, Germany, AACE. - pp. 1242-1247.

Schwarz, E. (1998). Self-organized goal-oriented tutoring in adaptive hypermedia environments. In B. P. Goettl, H. M. Halff, C. L. Redfield, & V. J. Shute (Ed.), *4th International Conference, ITS-98* (pp. 294-303). Berlin: Springer-Verlag.

Specht, M., & Klemke, R. (2001). ALE - Adaptive Learning Environment. In W. Fowler, & J. Hasebrook (Eds.), *Proceedings of WebNet'2001, World Conference of the WWW and Internet,* October 23-27, 2001. Orlando, FL, AACE. - pp. 1155-1160.

Specht, M., Kravcik, M., Klemke, R., Pesin, L., & Hüttenhain, R. (2002). Adaptive Learning Environment (ALE) for Teaching and Learning in WINDS. In *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002)* (Vol. 2347, pp. 572-581). Berlin: Springer-Verlag.

Specht, M., & Oppermann, R. (1998). ACE - Adaptive Courseware Environment. *The New Review of Hypermedia and Multimedia*, **4** 141-161.

Specht, M., Weber, G., Heitmeyer, S., & Schöch, V. (1997). AST: Adaptive WWW-Courseware for Statistics. In P. Brusilovsky, J. Fink, & J. Kay (Eds.), *Proceedings of Workshop "Adaptive Systems and User Modeling on the World Wide Web" at 6th International Conference on User Modeling, UM97,* June 2, 1997. Chia Laguna, Sardinia, Italy, Carnegie Mellon Online. - pp. 91-95, available online at http://www.contrib.andrew.cmu.edu/~plb/UM97_workshop/Specht.html.

Steinacker, A., Faatz, A., Seeberg, C., Rimac, I., Hörmann, S., Saddik, A. E., & Steinmetz, R. (2001). MediBook: Combining semantic networks with metadata for learning resources to build a Web based learning system. *Proceedings of ED-MEDIA'2001 - World Conference on Educational Multimedia, Hypermedia and Telecommunications,* June 25-30, 2001. Tampere, Finland, AACE. - pp. 1790-1795.

Steinacker, A., Seeberg, C., Rechenberger, K., Fischer, S., & Steinmetz, R. (1999). Dynamically generated tables of contents as guided tours in adaptive hypermedia systems.Educational Multimedia/Hypermedia and Telecommunications, 1998, *Proceedings of ED-MEDIA/ED-TELECOM'99 - 11th World Conference on Educational Multimedia and Hypermedia and World Conference on Educational Telecommunications,* Seattle, WA, AACE. - pp. .

Stern, M. K., & Woolf, B. P. (2000). Adaptive content in an online lecture system. In P. Brusilovsky, O. Stock, & C. Strapparava (Ed.), *Adaptive Hypermedia and Adaptive Web-based systens* (pp. 225-238). Berlin: Springer-Verlag.

Süß, C., Kammerl, R., & Freitag, B. (2000). A teachware management framework for multiple teaching strategies. In J. Bordeau, & R. Heller (Eds.), Educational Multimedia/Hypermedia and Telecommunications, 1998, *Proceedings of ED-MEDIA'2000 - World Conference on Educational Multimedia, Hypermedia and Telecommunications,* June 26 - July 1, 2000. Montréal, Canada, AACE. - pp. ??

Trella, M., Conejo, R., & Guzmán, E. (2000). A Web-based socratic tutor for trees recognition. In P. Brusilovsky, O. Stock, & C. Strapparava (Ed.), *Adaptive Hypermedia and Adaptive Web-based Systems, AH2000* (pp. 239-249). Berlin: Springer-Verlag.

Vassileva, J. (1990). An architecture and methodology for creating a domain-independent, plan-based intelligent tutoring system. *Educational and Training Technology International*, **27** (4), 386-397.

Vassileva, J. (1992). Dynamic CAL-courseware generation within an ITS-shell architecture. In I. Tomek (Ed.), *4th International Conference, ICCAL'92* (Vol. 602, pp. 581-591). Berlin: Springer-Verlag.

Vassileva, J. (1998). DCG + GTE: Dynamic Courseware Generation with Teaching Expertise. *Instructional Science*, **26** (3/4), 317-332.

Vassileva, J., & Deters, R. (1998). Dynamic courseware generation on the WWW. *British Journal of Educational Technology*, **29** (1), 5-14.

Verhoeven, B., Cardinaels, K., Van Durm, R., Duval, E., & Olivié, H. (2001). Experiences with the ARIADNE pedagogical document repository. *Proceedings of ED-MEDIA'2001 - World Conference on Educational Multimedia, Hypermedia and Telecommunications,* June 25-30, 2001. Tampere, Finland, AACE. - pp. 1949-1954.

Weber, G., & Brusilovsky, P. (2001a). ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education*, **12** (4), 351-384, available online at http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/weber.html.

Weber, G., Kuhl, H.-C., & Weibelzahl, S. (2001b). Developing adaptive internet based courses with the authoring system NetCoach. In P. D. Bra, P. Brusilovsky, & A. Kobsa (Eds.), *Proceedings of Third workshop on Adaptive Hypertext and Hypermedia,* July 14, 2001. Sonthofen, Germany, Technical University Eindhoven. - pp. 35-48, available online at http://wwwis.win.tue.nl/ah2001/papers/GWeber-UM01.pdf.

Weber, G., & Specht, M. (1997). User modeling and adaptive navigation support in WWW-based tutoring systems. In A. Jameson, C. Paris, & C. Tasso (Ed.), *6th International Conference on User Modeling* (pp. 289-300). Wien: SpringerWienNewYork, available online at http://www.psychologie.uni-trier.de:8000/projects/ELM/Papers/UM97-WEBER.html.

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