Proficiency-Adapted Information Browsing and Filtering in Hypermedia Educational Systems

LICIA CALVI
Department of Romance Languages and Literatures
University of Antwerp (UIA)
Antwerp, Belgium
e-mail: calvi@uia.ua.ac.be

PAUL DE BRA
Department of Computing Science
Eindhoven University of Technology
Eindhoven, The Netherlands
e-mail: debra@win.tue.nl

Abstract. We present a framework for proficiency-adapted information browsing and filtering in educational hypermedia systems. In hyperdocuments, information is acquired by browsing through highly interconnected sets of information nodes. In order to find specific information, users follow links to nodes they judge to be relevant. In order to help users find relevant information and new learning material that match their levels of domain knowledge, we present a framework for adapting the information nodes, and the links leading to them, to the user's *proficiency* in the subject matter. Such a proficiency-adapted, user-centered educational environment is intended to enhance learning.

We believe that learning in educational hypertext-based applications cannot be reduced to traversing a static information space. Navigating through any space, be it a physical or an information space, normally requires that users have a prior degree of proficiency in the domain knowledge. Learning is an evolving dynamic process through which users progress from a situation of unfamiliarity to one of mastery of a knowledge corpus. Therefore, we propose a model of proficiency-adapted learning and information browsing in which the presented choices (links and the textual context of links) are selected based on the user's knowledge state. Ultimately, such an adaptive course not only guides the learning process of the student, but it gradually transforms itself into a reference guide.

Key words: user-adaptivity, adaptive hypermedia systems, intelligent tutoring, information browsing, information filtering, cognitive load, self-modifying hyperdocuments.

1. Introduction

Since World Wide Web is increasingly becoming the preferred way for information provision, it is necessary that its information (or hypermedia information in general)

is presented in a form that meets users' (learning) requirements and suits their (cognitive) abilities. This is not as trivial as it may seem at first glance since it ultimately entails an analysis of differences in human cognitive styles which concern *how* information is processed (Messick, 1976). It also requires a specification of short-term memory capacities (Miller, 1991), concerning instead *how much* information can eventually be processed. One parameter in the latter context is the user's *cognitive load* that will be discussed below.

Findings in the instructional sciences and writing research suggest that the way in which information is *structured* is of paramount importance. One may safely assume that this also holds for hypermedia educational systems. Even though what the user learns is the content of information, and not its structure, the way information is presented (and thus visually structured) can strongly enhance its comprehension and consequently its learning. This is particularly true for educational systems, where learning is conceived as the progression from a knowledge state A to a state B, i.e., from a situation of unfamiliarity to one of mastery of the knowledge corpus. In this paper, we present a framework for adapting the links and the textual context of links in hyperdocuments to the user's *proficiency* in the subject matter, in order to help users comprehend and learn the contents of these documents. An example adaptive hypermedia course, namely the course "2L670: Hypermedia Structures and Systems" (2L670), will be presented, and some theoretical issues concerning comprehension and learning will be discussed. This adaptive hypermedia course has been developed at the Eindhoven University of Technology and is offered at six universities in the Netherlands and in Belgium (through distance education by means of Internet).

Research on adaptive hypermedia systems is fairly new (see Brusilovsky 1996 for a survey). These systems often employ complex user modeling techniques and/or dynamic content and link structure generation. We try a simple approach in which only the content and the links for which the user is ready are enabled, while the remaining links are temporarily disabled or even removed. While information retrieval and learning through browsing is already less demanding for users than more traditional querying, adapting the browsing space by restricting it based on users' proficiency makes browsing even easier since only those collections of nodes that form meaningful new learning units are presented, and not *all* information clusters that are related. This process of adapting to users' evolving knowledge treats navigation as progress rather than merely as traversal.

This paper is organized as follows: Section 2 presents the general context of the problem. Section 3 introduces the issue of cognitive load and the need to design hyperdocuments for comprehension. Section 4 illustrates the proposed adaptive model of information seeking in light of a constructivistic theory of learning. Our framework has been applied to the courseware "2L670: Hypermedia Structures and Systems". This application is described (Section 5), discussed (Section 6), evaluated (Section 7), and compared with other approaches (Section 8).

2. Problem Definition

> *"It may be fun and perhaps instructive to open every door and peek inside, but there are many situations where learning is more effective when the freedom of the learner is restricted to a relevant and helpful subset of activities"* (Hammond, 1993)

In educational contexts, pure navigation-based systems have been criticized because users tend to navigate through the knowledge corpus without any pedagogical guidance: they simply follow the links which they find interesting or attractive. Even though this is true for many other software systems as well, it is particularly problematic with educational systems where users are supposed to gain familiarity with the system and its content by navigating and reading. Most existing navigational support tools are not very helpful in the primary learning phases since they always implicitly rely on some form of domain knowledge on the part of the users which, in the initial stage, these users normally lack. Shifting learning control from the educator to the learner puts a mental load on users. Whether this is an advantage or disadvantage for novice users who are not aware of the basic prerequisites of the domain structure and of cognitive self-control in spatial representations of knowledge is still an open question (Steinberg, 1989, Linard, 1993). One possible way of alleviating such a potential pitfall is to provide some device to filter links and thus to restrain link selection by means of *guidance*.

A common problem with static hypertext is indeed the abundance of links. The creation of links is by far the most difficult aspect of hypertext authoring. The decision whether or not to create a link between node A and node B is influenced by several issues, like, e.g.:

1. Is the reader of node A capable of understanding the information in node B, irrespective of the path she followed to reach node A?
2. Does node B provide interesting (new) information that a reader of node A might want, irrespective of the path she followed to reach node A?
3. Will switching from node A to B disturb the reading process for the content of node A?

In many cases it is difficult, if not impossible, to answer these questions because of the phrase "irrespective of the path she followed to reach node A". What is often done is therefore to create many links, keeping in mind the first part of the first question ("Can the reader understand node B"), but ignoring the first part of the second question ("Does node B provide interesting new information"). Unfortunately such a rich link structure is often a navigational nightmare.

In both InterBook (Brusilovsky *et al*., 1996a) and ELM-ART (Brusilovsky *et al*., 1996b) an *adaptive link annotation* formalism has been introduced as an attempt to provide guidance through the rich link structure. But DeYoung (1990) suggests that even such an annotation may not be sufficient:

> *"(...) if a hypertext is a densely connected unstructured graph, no presentation of the mess is going to clean it up or make it conceptually easier to understand"* (DeYoung, 1990).

3. Cognitive Models for Structuring Hypermedia: Designing for Comprehension

Due to the extra navigational freedom which they provide, hyperdocuments impose greater cognitive loads on users than linear documents, such as books or reports on paper or on-line. As a consequence, when the subject being taught is relatively simple and well structured, enough short-term memory processing capacities are left for the learning task. But when the learning task is more complex, it becomes critical that information is presented in an easily accessible way (Riding and Rayner, 1995). Otherwise, the probability of important information not being learned properly because there has been no actual comprehension increases dramatically.

In the cognitive science tradition, comprehension indicates the user's ability to build a mental model representing the objects and the semantic relations implicit in the document the user is analyzing (van Dijk and Kintsch, 1983). And, in the case of hyperdocuments, one of the truisms in HCI states that users build some form of mental model of the device and structure they are interacting with in order to understand the system's application domain.

These statements entail two important consequences:

1. The readability of a document is a function of the cognitive effort spent on the construction process. A readable document requires little cognitive overhead.
2. By contrast, the clearer the model that the user is able to form, the more capable she will be at grasping the application domain (Calvi and Geerts, 1995).

Therefore, in order to enhance the readability of a hyperdocument, users have to be assisted in constructing their mental models by strengthening those factors supporting this process and, conversely, by weakening those that impede it.

We focus in particular on two factors that affect this construction process: *coherence* (Thüring *et al*., 1991) and *cognitive overhead* (Conklin, 1987). Coherence affects the comprehension dynamics in a positive way. The user's capacity to understand and to remember a document depends on the degree of coherence of a document, which is related to the user's ability to construct a mental model corresponding to facts and relations in the subject domain of the document (Johnson-Laird, 1983). Coherence can be *local* , i.e., within elements of the subject domain, or *global,* i.e., between subject domains. In this way users can construct a macrostructure subsuming all pieces of information (van Dijk and Kintsch, 1983).

In a linear (book-like) document, both local and global coherence can be achieved by making the document well-structured and providing rhetorical cues which mirror this structure. In a hyperdocument this is not so simple. If we see the local coherence to be on the node level, i.e. within hyperdocument pages, and the global coherence to be between nodes, i.e. between hyperdocument pages, then system designers have to *"limit the fragmentation characteristic of hypertext"* (Marshall

and Irish, 1989). Fragmentation seems to be especially problematic in hyperdocuments since information is segmented into disjoint nodes, frequently sequestered by means of different representation media, and spread over several windows. As a consequence of this sequestering, users may experience a lack of interpretative context which can give them the impression that the hyperdocument is just an aggregation of loosely linked pieces of information, and not a coherent whole.

In contrast to coherence, cognitive overhead impairs comprehension. Cognitive overhead is *"(...) the additional effort and concentration necessary to maintain several tasks or trails at one time"* (Conklin, 1987) and *results from* intrinsic human constraints in information processing (Kahneman, 1973). Every task on top of mere reading requires yet another set of cognitive skills and takes cognitive bandwidth, thereby reducing the mental resources available for comprehension.

4. A Model for Proficiency-Adapted Information Browsing and Filtering

Ultimately, hyperdocument structures for easy comprehension and effective learning must take structures of learning into account, i.e., cognitive models within the learner (Eklund, 1995). This view of learning entails the assumption that students actively build up knowledge on the basis of their previous understanding. They do so by *constructing* it from the content of a document in a self-motivated and self-directed fashion and alter existing structures through the addition and deletion of links and nodes.

4.1. THE LEARNING PARADIGM

Chan *et al.* (1993) map such a constructivistic model of learning into the following four learning stages:

I. an *orientation* phase, through which the learner acquires new knowledge based on previous understanding by dynamically interacting with the learning media;
II. a *coaching* phase, also defined as apprenticeship learning;
III. a *tuning* phase, to acquire a more specific insight into the new data, and;
IV. a *routinization* phase, in which the learner gradually becomes more and more autonomous with respect to the old knowledge, thanks to the experience gained.

Thus the initial learning phase cannot be based on domain representations because they are supposed to be the final outcome of the learning process, and hence not yet available to students when starting the course. At this stage, users can only rely on their previous knowledge.

If we think of the four learning phases in terms of a cognitive model for structuring hypermedia, we can recognize the influence of the following processes:

I. *connection*: where weak links are created between the prior knowledge and the new material;
II. *accretion*: where the newly acquired knowledge is further expanded by creating additional weak links;
III. *articulation*: which causes some links to be strengthened and others to be (simultaneously) deleted;
IV. *solidification*: where the remaining links and their connected units are further strengthened.

Thus, learning can be viewed as the accumulation and subsequent organization of knowledge structures, which are represented in terms of links and nodes (see Fig. 1). While the nodes are declarative elements of knowledge defined as clusters of episodes (i.e., as small units of related material from the content domain), links represent their procedural and structural understanding. Through repetition, and then through experience, this node-link dyad becomes just a node, i.e., an element of the declarative knowledge. This alters the knowledge granularity within the learner's understanding (Eklund, 1995).

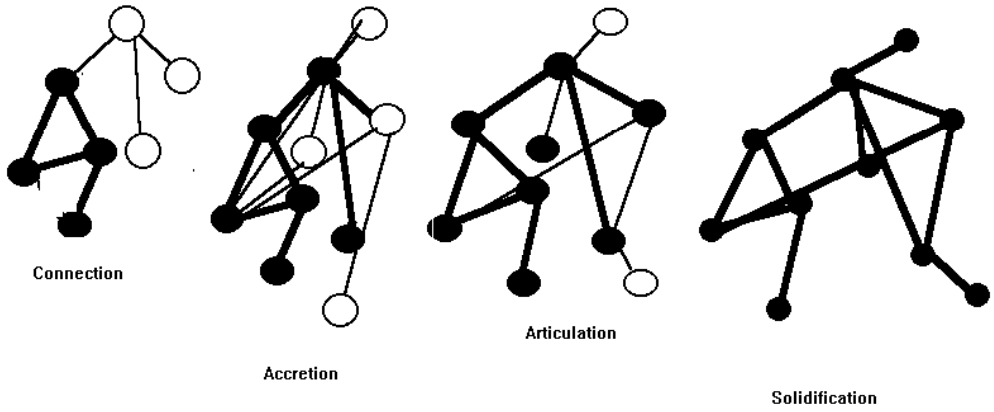

Figure 1. The four-stage model of learning progress (from Chan et al., 1993).

Following this theory, learning with hypermedia systems can be enhanced by integrating model structures representing the learner's knowledge acquired from these systems (Jonassen, 1991) and taking this model into account when presenting information to the user.

5. An Example Adaptive Tutoring System: The Courseware "2L670"

In this section we will show in detail how the theoretical model discussed so far has been successfully applied to the design of the courseware "2L670: Hypermedia Structures and Systems", which has been developed at the Eindhoven University of Technology and which is currently offered to students at six different universities in the Netherlands and Belgium through the World Wide Web. The course was first

taught in 1994. This early courseware was still based on a static link structure. Experience with it has led to the development of new courseware, which is equipped with a dynamically changing link structure and dynamic content presentation, depending on each student's proficiency in the subject matter.

5.1. ADAPTIVE HYPERMEDIA ON THE WORLD WIDE WEB

In the area of computer-based learning, the combination of hypermedia techniques and World Wide Web technology has so far been mostly applied to computer science courses where students were asked to master certain skills, such as programming. Examples include tutorial systems for programming in C (Kay and Kummerfeld 1994a; 1994b) and for programming in Lisp (Brusilovsky *et al*., 1996a), and the CyberProf platform (Hubler and Assad, 1995) in the areas of on-line homework and quizzes. In all cases, the hyperstructure offered by the World Wide Web was not fully exploited because the course material of all above-mentioned prototypes is mostly hierarchical. This is due to the fact that the learning required by these systems (i.e., learning a programming language and solving course problems), is considered a mostly linear process. Indicating which chapters are still to be avoided and which pages are to be read first is relatively straightforward, resulting in a fairly simple (link) structure.

Authoring adaptive hypermedia in a Web environment requires additional considerations because of the lack of a "conditional text" command in the HyperText Markup Language (HTML). This makes it impossible to create HTML documents of which only selected parts are presented to the user on the basis of environment variables which are ultimately controlled by an agent that monitors the user's knowledge state or preferences. Some attempts have already been made to circumvent this limitation. For instance, the InterBook system (Brusilovsky *et al.,* 1996a) uses *concept-based indexing* to provide access to (non-adaptive) HTML pages from dynamically generated index pages. There, index and "content" pages are displayed simultaneously by means of *frames*, a technique introduced by Netscape which is not available in all browsers and not part of the current HTML 3.2 standard. Developers who use programmable WWW servers (Weber and Specht 1997) and middleware (Fink *et al.,* 1996) to dynamically generate user-tailored WWW pages therefore employ structures other than HTML to represent conditional content.

In the courseware for "2L670: Hypermedia Structures and Systems", conditional text is encoded in (standard) HTML documents (De Bra and Calvi, 1997). This new approach supersedes a first attempt at providing adaptive content, as described in (Calvi and De Bra, 1997). There, much like in (Van Ginderen, 1990), we had enhanced standard HTML by **#if** constructs and used a filter, namely the Unix C pre-processor, to select between arbitrary pieces of text. (In Section 5.4 we show a newer approach towards providing adaptive content, without using the Unix C pre-processor.) Although this option offers the possibility to make large modifications, in that particular version of the course we were using adaptive presentation for small variations in content only, like, e.g., in the table of contents and in some other places, where additional text was added to explain why certain chapters were still

unavailable, and in the overall text, where some technical terms, like *node* and *link,* were carefully replaced by non-technical terms like *page* and *reference* or *cross-reference*.

In addition to adaptive presentation, the course also provides adaptive navigation support by means of adaptive enabling and disabling of links. The idea here is that only links to pages which are supposed to be "interesting" for the student to read next are made accessible, while the "premature" links remain disabled and invisible until an advancement in the student's learning progress determines a shift in their relative importance. (In our next version of the courseware we are shifting towards *link hiding* which means that "premature" links are accessible but invisible, i.e., indistinguishable from normal text. We will investigate the differences between the two approaches.)

5.2. MODELING THE STUDENT: ADAPTING THE LINK STRUCTURE TO THE STUDENT'S PROFILE

In several adaptive hypermedia systems, users are classified as belonging to one of the categories "novice", "advanced", or "expert". On the basis of a pre-established learning plan, each of these categories is associated with a well-defined instructional path, i.e., a navigational path through the hypermedia learning material. Each achievement of the learner (in terms of read nodes and still-to-be-read nodes) is stored separately in an individual user logfile, and is used to determine whether a user is a novice or an expert. More advanced adaptive hypermedia systems use a more detailed description of the user's level, based on all the information in the user's logfile. The complete navigation path the user has taken to reach the "current" page can influence the presentation and structure of that page and of the whole hyperdocument at any time in the future.

The course "2L670: Hypermedia Structures and Systems" is an example of an adaptive hypermedia system that takes the complete browsing history into account. When the student first visits the course, the starting page displays a kind of table of contents, but only a single link is enabled (namely to a node containing instructions about the use of the courseware). This enforces a single initial navigation path for all users. After reading the instruction page, the first three "chapters" of the course text become accessible, which contain an introduction, a set of definitions, and a historic overview of hypermedia research and development. Each student may follow a different path through these chapters, and depending on the path taken, the link structure within these chapters may be adapted differently. Once the student has read some of the nodes of these chapters, and completed a small multiple-choice test at the end of each chapter, links to the more advanced chapters appear in the above-mentioned table of contents, thus increasing the navigational freedom. While reading, some links will also become deactivated again, like e.g. the links to definitions of terms or to nodes containing unimportant background information (such as the nodes describing the IBM PC or the Macintosh computer, which we do not expect students to wish to read multiple times). Deactivated (or de-emphasized in the latest version of the courseware) links try to compensate for the links that are added, preventing the hyperdocument from becoming a mess of abundant cross-

references. By keeping track of the concepts learned by the user (and represented through *knowledge identifications* (id's)), we can make links accessible when the user is ready for them (according to our constructivistic model design), and make them inaccessible again after the user has no need for them anymore. The link structure of the course text thus changes each time the student follows a link and, consequently, the essence of the courseware changes accordingly. Eventually, once the student masters the subject, the course text will have modified itself into a reference manual for experts (see below).

5. 3. ENCODING STUDENT'S KNOWLEDGE

The course "2L670" is divided into "chapters" which are listed on its first page containing the table of contents. As mentioned in the previous section, the course "2L670" contains some introductory chapters, giving definitions and a historical overview, and advanced chapters, describing reference models, navigation and retrieval problems, authoring issues and multi-user aspects. It is desirable for students to first read the introductory chapters, and therefore to advise or force them to do so (by dimming, disabling or removing links to the advanced chapters at first).

Chapters are however overlapping sets of pages. For a number of pages it is actually impossible to tell to which unique chapter they belong. Even an introductory chapter and an advanced chapter may share common pages. This renders enabling or disabling access to pages more complicated than simply enabling or disabling whole chapters, and it also suggests a "simple" wording when a page is read as part of an introductory chapter, and more technical wording when the same page is read as part of an advanced chapter.

In order to monitor the student's progress and knowledge proficiency, *concepts* are associated with pages from the course text. Each concept is denoted by means of a single word or compound. Much like in De Rosis et al. (1994), the concepts are collected in a concept dictionary. While for programming languages and similar courses the user model needs to consist of both "KNOW-ABOUT" and "PRACTICE-IN" facts, we currently make no such distinction. The knowledge proficiency of a student is simply a set of concepts the student has read about (or successfully taken a test about). "KNOW_ABOUT" and "PRACTICE-IN" can of course be simulated by replacing concepts by pairs.

For each page of the course text, a number of concepts may be prerequisite knowledge and a number of other concepts may make the page redundant. In (Calvi and De Bra, 1997) this prerequisite and/or forbidden knowledge was used to determine whether to enable or disable links to a page. More complex Boolean combinations were not possible in that proposal. There, the software used in the course "2L670" could keep track of all the nodes read by each individual student and associate to each node three "knowledge identifications", namely:

- ***Required knowledge,*** i.e., the knowledge that is needed to view the current node. The system will automatically disable conditional links to nodes for which the user has not yet acquired this required knowledge.

- ***Forbidden knowledge,*** i.e., the knowledge that makes viewing the current node uninteresting. The system will automatically disable conditional links to nodes for which the user has already acquired this knowledge.
- ***Generated knowledge,*** i.e., the knowledge the user gains after reading the current node.

Whether or not conditional links appear depends on the combined knowledge id's a student has acquired. In the course "2L670" there are also *unconditional links* which are always shown, regardless of the student's knowledge. In the latest version of the courseware "required" and "forbidden" knowledge have been generalized to an arbitrary Boolean expression using concepts.

5.4. ADAPTIVE CONTENT PRESENTATION

In adaptive hypermedia systems, the organization of content (and possibly also the content itself) changes over time, based on the (past) browsing strategy of the user on the one hand and the knowledge representation on the other hand. Both refer to what Eklund and Zeiliger (1996) call *external structure* of adaptive hypermedia systems: the organization of content into nodes and links is dynamically adapted to match the user's specific needs and learning progress.

In (Calvi and De Bra, 1997) we had proposed to use C pre-processor constructs (**#ifdef**) to achieve this goal (see Section 5.1). However, mixing HTML with C pre-processor statements makes authoring unnecessarily complicated. Since we aim to provide an authoring environment which is also suited for the development of non computer science courses, authoring needs to be kept simple and intuitive. Also, we want authors to be able to use standard HTML editors or word processors that can generate HTML. In (De Bra and Calvi, 1997), we use HTML comments to mix "if-statements" with HTML text, and use a filter to select the appropriate pieces of text depending on the concepts the student has mastered. Whole pages, links to pages and pieces of HTML text (possibly including images), can be enabled or hidden depending on a Boolean combination of concepts.

The following example of source text shows how the adaptive index page for the hypermedia course text might look like:

```
<!-- requires true -->
<!-- generates nothing -->
    <h1>Hypermedia structures and systems</h1>
    Welcome to course 2L670 at the Eindhoven University of Technology.<p>
<!-- if not readme -->
    Since you are just beginning to browse through this course,  you should first read
    <a href="readme.html">the instructions</A>.  These will explain how to use this course text.
    In order to get to the instructions you must click (the left or only mouse button) on the phrase
    <a href="readme.html">"the instructions"</a>.<p>
    The items below indicate (not necessarily disjoint) parts of the course text, which will
```

```
        become accessible after you have read <a href="readme.html">the instructions</a>.<p>
    <!-- else -->
        This course contains the following (not necessarily disjoint) parts:
    <!-- endif -->
        <ul>
        <li><a href="intro.html">Introduction</a> (it is advised to read this before the other items)
        <li><a href="definition.html">Definition of hypertext and hypermedia</a>
        <li>The <a href="history.html">history</a> of hypertext and hypermedia
    <!-- if readme but not (introduction and definition and history) -->
        </ul>
        The following parts will become available later (when you are ready for them):
        </ul>
    <!-- endif -->
        <li>The <a href="architecture.html">architecture</a> of hypertext systems
        <li><a href="navigation.html">Navigation</a> (and browsing semantics) in hypertext
        <li><a href="retrieval.html">Information Retrieval</a> using hypertext
        <li><a href="authoring.html">Writing</a> hypertext
    <li><a href="distribution.html">Distribution and Concurrency</a> issues
        <li><a href="future.html">The Future of Hypertext and Hypermedia</a>
        <li><a href="assignment.html">Assignment for this course</a>
        </ul>
```

5.5. CONDITIONAL TEXT IN HTML

Each page of the course material starts with an HTML comment that indicates which concepts are *required* or *forbidden* for accessing this page. In the above example, this condition is set to “true” (and the page therefore always accessible), but it may as well be an arbitrary Boolean expression using concepts. The second HTML comment indicates which concepts are *generated* after visiting the page, namely the concept “nothing” in this example (we would normally omit this clause when no new knowledge is generated). The generated concepts are added to the student’s knowledge **after** reading the page. This implies that text fragments that depend on the concepts generated by a page are not displayed the first time the page is visited. It also implies that a page may forbid the same concept(s) it generates, in which case links to this page will be shown only once.

Note that although in the example the links to all chapters are always present in the source text, a second filter (De Bra and Calvi, 1997) will remove or disable these links until the student has gained sufficient knowledge. This filter was added to the CGI-script that maintains each student’s individual log file (De Bra, 1996). The filter currently distinguishes between *conditional* links (which only appear when considered appropriate) and *unconditional* links (which are always shown) based on the use of upper- or lowercase in the word “HREF” in the HTML anchor tag. Instead the newer version of the courseware will use link classes using the “CLASS” attribute of the HTML anchor tag. Although the upper- vs. lowercase trick was easy, some HTML editors do not give the author control of the case used for the “HREF” field (or any other field). Instead of relying on a special link filter we could have used more **<!-- if -->** commands (actually comments in HTML) to include these links conditionally, but that would make the document source much harder to write (and read). Note also that the keyword *but* in the second **<!-- if -->** statement is simply used as a synonym for *and*, but is closer to natural language.

When the student first looks at this page, the following text will be presented:

Hypermedia structures and systems

Welcome to course 2L670 at the Eindhoven University of Technology.

Since you are just beginning to browse through this course, you should first read the instructions. These will explain how to use this course text. In order to get to the instructions you must click (the left or only mouse button) on the phrase "the instructions".

The items below indicate (not necessarily disjoint) parts of the course text, which will become accessible after you have read the instructions.

- Introduction (it is advised to read this before the other items)
- Definition of hypertext and hypermedia
- The history of hypertext and hypermedia
- The architecture of hypertext systems
- Navigation (and browsing semantics) in hypertext
- Information Retrieval using hypertext
- Writing hypertext
- Distribution and Concurrency issues
- The Future of Hypertext and Hypermedia
- Assignment for this course

After reading "the instructions", the page looks as follows:

Hypermedia structures and systems

Welcome to course 2L670 at the Eindhoven University of Technology.

This course contains the following (not necessarily disjoint) parts:

- Introduction (it is advised to read this before the other items)
- Definition of hypertext and hypermedia
- The history of hypertext and hypermedia

The following parts will become available later (when you are ready for them):

- The architecture of hypertext systems
- Navigation (and browsing semantics) in hypertext
- Information Retrieval using hypertext
- Writing hypertext
- Distribution and Concurrency issues
- The Future of Hypertext and Hypermedia
- Assignment for this course

Finally, after also reading (some pages of) the first three chapters, the presentation changes to:

Hypermedia structures and systems

Welcome to course 2L670 at the Eindhoven University of Technology.

This course contains the following (not necessarily disjoint) parts:

- Introduction (it is advised to read this before the other items)
- Definition of hypertext and hypermedia
- The history of hypertext and hypermedia
- The architecture of hypertext systems
- Navigation (and browsing semantics) in hypertext
- Information Retrieval using hypertext
- Writing hypertext
- Distribution and Concurrency issues
- The Future of Hypertext and Hypermedia
- Assignment for this course

5.6. VERIFYING ADAPTIVE LINK STRUCTURES

In static hyperdocuments, analyzing whether all nodes can be reached from a given "root" node is a matter of simple graph traversal. The Boolean conditions however complicate the link structure, because links may or may not be present depending on the student's knowledge state. As long as no negation is used in preconditions, the *reachability problem* is still easy: one can reiterate the forward traversal each time some knowledge is gained. Fig. 2 illustrates a case of unreachable pages: no pages other than the root node can be accessed because the pages that generate concepts can not be reached without passing through pages that require that knowledge. Reachability is therefore a necessary condition for readability.

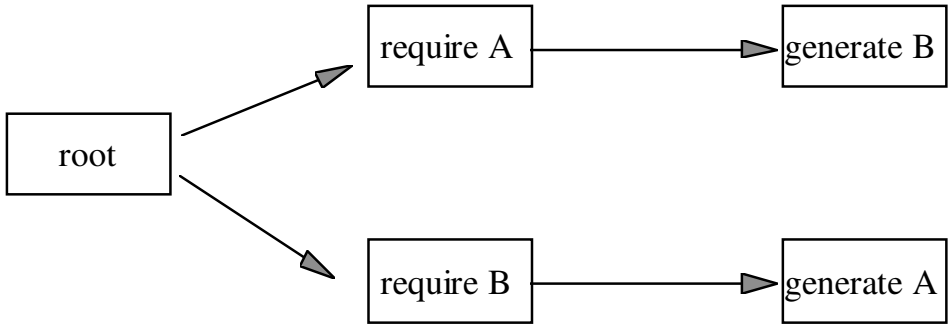

Figure 2: Unreachability due to unfulfillable preconditions

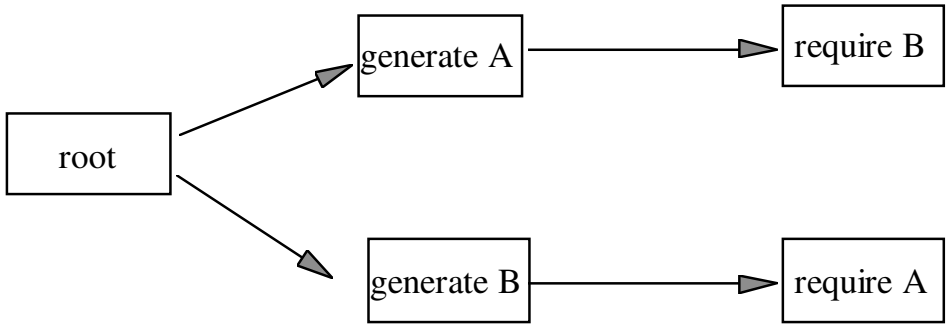

Figure 3: Difficult reachability

Reachability in itself is however not a sufficient condition for *easy* readability. Fig. 3 illustrates an undesirable construct: in order to reach the nodes that require knowledge about A resp. B, the student must visit the node that generates A, then go back to the root, then visit the node that generates B, then the node that requires A, then go back to the root, revisit the node that generates A, and then visit the node that requires B. An "easy" link structure would not require the student to do this. A quality measure for link structures is the accessibility of each node without extensive page revisits.

Negation in preconditions complicate matters even further since the gain of knowledge may make links unavailable. In the above table of contents page for the hypermedia course, the link to the "instructions" page disappears *after* reading the instructions (the page can still be reached through a complete index, as described in De Bra, 1996). In this case, the disappearing link does not make reading a page impossible because the link deletion happens *after* the student has read the instructions. However, in general it is possible that by reading pages in a peculiar order some pages may never become available to the student. The quality measure for link structure is the absence of navigation paths that leave some pages

inaccessible forever. Fig. 4 illustrates such a case: the page that depends on neither knowing concept A nor B can never be reached.

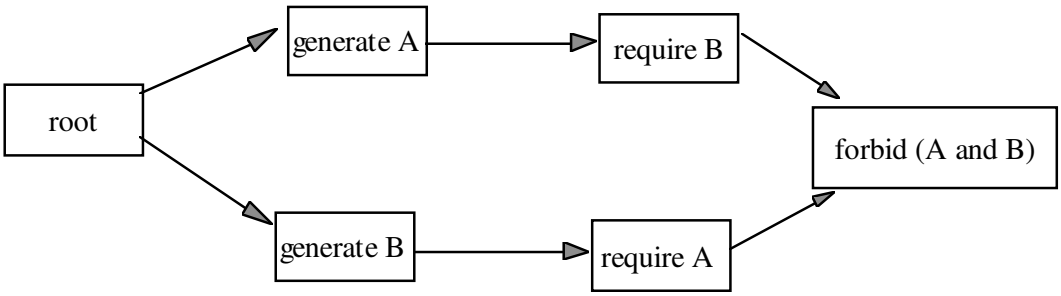

Figure 4: Unreachability due to unfulfillable negative preconditions

In order to avoid such occurrences, we are currently building a simple tool that verifies the link consistency of a course text. In particular the tool analyzes the following properties:

- Are there navigation paths that make it impossible to visit some page(s)? If so, which pages may not be reachable? (This can be a consequence of using negative preconditions, but also of requiring concepts which are never generated before they are needed).
- Are there (conditional) parts of pages that can never be viewed (no matter which navigation path is used)?
- Is it possible to navigate through the whole course text without ever following a forward link to a page that was visited before? (Given a course text like "2L670", this implies that it must be possible to read the text chapter by chapter). If not, which pages must be revisited in order to gain access to which pages?

6. Discussion

So far we have tried to emphasize that the current adaptive courseware, with links being created and removed at different points throughout the learning process, and content being modified accordingly, uses a constructivistic, proficiency-based metaphor. However, this kind of adaptive behavior can also be used to gradually convert the hyperdocument to a very different one. A course is originally delivered as a textbook in the form of an on-line hyperdocument. The reading order is neither intended nor expected to be linear as for paper-based textbooks, but nevertheless introductory material must be read and mastered before more advanced topics can be learned. Adaptive hyperdocuments can restrict the navigational freedom of beginning students and thereby enforce a pedagogically meaningful learning sequence. When the student's proficiency increases, a different, partially

hierarchical structure emerges, which gives easy access to all information through short navigation paths from the start page (e.g., the table of contents). This final structure is more typical for encyclopedic work or reference guides. The courseware is then no longer suitable as textbook, but only as a reference book, where the specialized concepts can be accessed immediately without having to go through the complete learning process every time. To achieve this, both the hyperdocument's link structure and the content and/or presentation need to be adaptive. Links that are present or absent often need a different textual context. In the course "2L670" we actually use adaptive content primarily to provide an appropriate textual context for links.

We do not claim that adaptive hypermedia has no drawbacks. If it is clear from the previous discussion that a dynamic link structure can be used to eliminate the confusion that could be caused by providing links to information the student is not yet ready for, or that the student has already read and may not wish to find new links to, it is also apparent that seeing a different selection of outgoing links each time a node is revisited might be equally or even more confusing. Therefore, the pages with a varying link structure must be limited and well chosen. But as such, an adaptive link structure is used to unify two complementary (hyper)books which are intimately very different: a course text for learning and a reference text for later use. This also implies that the user typology has modified itself in-between: from novice to expert user.

7. Preliminary Evaluation

The adaptive courseware has been used in a trial in 1996, before becoming the standard version as of January 1997. Organized user surveys still have to be completed (by comparing the log files from each student's learning process), but we can already draw some conclusions from informal meetings with some of the students (from two out of the six universities). In general, users feel uncomfortable about links being removed because they know there is more material they can ultimately visit but they do not know how to reach it. This is in line with the empirical results of Specht (1997) that guidance through link deactivation (in combination with link annotation), but *not* through link removal, significantly reduces the reading time and increases the learning effect in comparison with static hypertext. In the next version of the course, links to unsuitable pages will therefore be *de-emphasized* rather than removed.

Eventually, the carefully designed adaptive link structure both increases and reduces the user's freedom in the following ways:

- When a student is uncertain whether links will lead to comprehensible information (false directions become soon apparent after following a few bad links), she will exhibit a navigational behavior that is close to a linear reading order. So, a rich static link structure would not lead to a free navigational behavior.

- While it is true that the navigational freedom in adaptive systems is restricted, since not all links are available at all times, the consequence of having only "safe" links available makes students less afraid to follow links that move away from what they perceive as being the "linear" reading order.

Users seem nevertheless to complain about the impossibility of the present formalism to provide them with a snapshot of the system's complete structure. Students indeed like to glance at the content of a textbook before actually starting to study the material. With the adaptive hypermedia course text this is not possible. With the (old) static version, in which all chapters become accessible right away, one can quickly look at a few nodes of each chapter, thus getting some idea what the course is about. However, this may not provide the same information as glancing through a book, because one does not get a rough impression of what is on all the pages but instead a detailed impression of what is on a few pages.

We believe that part of the problem with adaptive linking stems from the fact that while a link is removed, its link text and the sentence(s) surrounding the link remain unchanged. This text often strongly suggest the presence of the link. More and better use of adaptive node contents could solve this problem, and is already being worked on for the next version of the course "2L670".

Another enhancement of the course that still needs to be implemented are control questions to verify whether students really learned the course material on the pages which they access. At the moment the course "2L670" simply takes this for granted and proficiency is measured in terms of page access only. Beaumont (1994) gives an excellent example how the result of quizzes can be exploited for subsequently guiding students in an adaptive hypertext on Anatomy.

8. Related Work

Many different definitions of and techniques for adaptive hypermedia exist. A good overview is presented in (Brusilovsky, 1996), where the author distinguishes between two main categories of adaptivity, namely *adaptive presentation*, which includes both the selection of different media depending on users' preferences and the adaptation of a document's textual content based on a user's knowledge state, and *adaptive navigation support,* where the (apparent or effective) link structure between the hyperdocument pages is changed. Within the context of adaptive navigation support, Brusilovsky identifies in particular *direct guidance* (suggesting a next best link), *adaptive sorting of links* (displaying a list of links from best to worst), *adaptive hiding of links* (hiding, disabling or removing of inappropriate links), *adaptive annotation of links* (marking links as appropriate or inappropriate but not deleting or disabling any), or *map adaptation* (changing graphical overviews of the link structure). The description of the kinds of adaptation in the course "2L670: Hypermedia Structures and Systems" in the present paper followed the terminology of this author, with the exception that Brusilovsky's general notion of link hiding has been subdivided into link hiding, link disabling and link removal. In the current version of the course links are removed in the sense that the link text may still be there, but it is not a link. The next version will have link hiding,

meaning that the link text looks just like plain (black) text, but it is still a link. Link disabling would mean that the link text looks like a link, but it doesn't lead to its destination because it is disabled. Link disabling may be very disturbing for users, because, like the way link removal is used in the table of contents page of course 2L670, it shows users which directions there are, but doesn't let them select any of these disabled directions. Link hiding has become possible on the Web through the use of Cascading Stylesheets, a recent addition to WWW technology and supported in Netscape's Communicator from version 4.0 onwards, and in Microsoft's Internet Explorer from version 3.0 onwards.

Many previous studies on adaptive systems have only focused on dynamically assembling information and presenting it according to the user's class and/or knowledge state without including the user's learning procedure in identifying which information to present to students (Hirashima *et al.*, 1997). They lack the notion of a model of human cognition that describes both problem solving and the acquisition of procedural skills. In our case, variations in the link structure are not only needed for modifying the access path to advanced reading material. In a textbook it is common to first define a concept or term, and then move on to the applications of that concept. In reference books, the opposite order is sometimes also used: throughout the text some concepts or terms are used, which the user may or may not know; links (or references) to the definition of these concepts or terms provide access to these definitions, so the reader who is not aware of an exact definition can easily get to it. Such cross-references are also used in textbooks, but in most cases for backward references (pointing to information to review if needed), and seldom for forward references (pointing to previously unread information). Hence, the modifications to the link structure that are needed to transform a textbook into a reference book are more intricate than simple "unveiling" of links when the reader is ready to follow them.

The use of adaptive link hiding and link removal in the framework of the World Wide Web is then what makes "2L670: Hypermedia Structures and Systems" unique as educational hypermedia system. Most other systems rely either on direct guidance (e.g., Brusilovsky, 1992a; 1992b; Brusilovsky and Pesin, 1994; Zeiliger, 1993), or on adaptive sorting (Tomek *et al.*, 1993; Reed *et al.*, 1995). Adaptive link disabling has already been used in the educational hypermedia system of Pérez et al. (1995a, 1995b), which is however not WWW-based and employs isolated index links rather than links within text.

9. Conclusion

In this paper we have proposed an alternative model for information browsing and filtering in educational hypermedia systems. We call it proficiency-adapted, because it assumes a way of sequencing information, by adapting the link structure and the content presentation of a hypermedia system, which is based on the student's degree of mastery in the subject matter. A constructivistic theory of learning forms the theoretical foundation for our framework.

We have proposed a simple authoring environment for adaptive hypermedia courseware which offers the following features:

I. The content of the pages of the course text is adaptive, as well as the link structure.
II. The adaptive documents are written in standard HTML, and can be authored using (some of the existing) HTML editors.
III. Pages of the course text, as well as tests and assignments that may be embedded in the course text, generate knowledge about *concepts*.
IV. Information items (ranging from words to large parts of documents) and links can be made dependent on Boolean combinations of concepts (using the operators *and, or, not* , and arbitrary parentheses).
V. A verification tools allows authors to check whether all the information in the course can be reached by a student, independent of the order in which the student decides to view pages. This problem is non-trivial since knowledge about certain concepts can make pages inaccessible.

We have also presented an example implementation of a hypermedia courseware in which both the link structure and the content presentation adapt themselves to the student's learning proficiency. While guiding the learning process of the student, the course text (potentially) gradually transforms itself from being a course text to being a reference guide.

Acknowledgements

We would like to thank the three anonymous reviewers who gave valuable comments on previous drafts of the paper, as well as Alfred Kobsa who helped clarify some more points during the editing phase.

References

2L670: '2L670: Hypermedia Structures and Systems'.
(URL: http://wwwis.win.tue.nl/2L670/)

Beaumont, I.: 1994, 'User Modeling in the Interactive Anatomy Tutoring System ANATOM-TUTOR'. *User Modling and User-Adapted Interaction* **4**(1), 21-45.

Brusilovsky, P.: 1992a, 'A Framework for Intelligent Knowledge Sequencing and Task Sequencing'. In: C. Frasson, G. Gauthier and G.I. McCalla (eds.): *Intelligent Tutoring Systems,* Springer-Verlag, Berlin, pp. 499-506.

Brusilovsky, P.: 1992b, 'Intelligent Tutor, Environment and Manual for Introductory Programming'. *Educational and Training Technology International* **29**(1), pp. 26-34.

Brusilovsky, P.: 1996, 'Methods and Techniques of Adaptive Hypermedia'. *User Modeling and User-Adapted Interaction* **6**, pp. 87-129.

(URL: http://www.contrib.andrew.cmu.edu/~plb/UMUAI.ps).

Brusilovsky, P. and Pesin, L.: 1994, 'ISIS-Tutor: An Adaptive Hypertext Learning Environment'. Proceedings of JCKBSE'94, Japanese-CIS Symposium on Knowledge-Based Software Engineering, pp. 83-87.

Brusilovsky, P., Schwarz, E. and Weber, G.: 1996a, 'A Tool for Developing Adaptive Electronic Textbooks on WWW'. Proceedings of the WebNet'96 Conference, pp. 64-69.
(URL: http://www.contrib.andrew.cmu.edu/~plb/WebNet96.html).

Brusilovsky, P., Schwarz, E. and Weber, G.: 1996b, 'ELM-ART: An Intelligent Tutoring System on World Wide Web'. *Lecture Notes in Computer Science*, **1086**, pp. 261-269.
(URL: http://www.contrib.andrew.cmu.edu/~plb/ITS96.html)

Calvi, L. and De Bra, P.: 1997, 'Using Dynamic Hypertext to Create Multi-Purpose Textbooks'. Proceedings of ED-MEDIA'97, pp. 230-236.

Calvi, L. and Geerts, W.: 1995, 'Hypermedia and Multimedia: The Role of Mental Models to Support Navigation'. Proceedings of the 3rd Annual Mid-Atlantic Human Factors Conference, pp. 135-138.

Chan, T., Lin, C., Lin, S. and Kuo, H.: 1993, 'OCTR: A Model of Learning Stages'. Proceedings of AI-ED '93.

Conklin, J.: 1987, 'Hypertext: An Introduction and Survey'. *IEEE Computer*. **20**(9), pp. 17-40.

De Bra, P.: 1996, 'Teaching Hypertext and Hypermedia Through the Web'. Proceedings of the WebNet'96 Conference, pp. 130-135.
(URL: http://wwwis.win.tue.nl/~debra/webnet96/index.html).

De Bra, P. and Calvi, L.: 1997, 'Creating Adaptive Hyperdocuments for and on the Web'. (to appear) Proceedings of the WebNet '97 Conference.
(URL: http://wwwis.win.tue.nl/~debra/webnet97/adaptive-paper.html).

DeYoung, L.: 1990, 'Linking Considered Harmful. Designing and Reading Hyperdocuments'. In: N. Streitz, A. Rizk and J. Andre (eds.): *Hypertext: Concepts, Systems and Applications*. *Proceedings of the First European Conference on Hypertext*, Cambridge University Press, pp. 238-249.

De Rosis, F., De Carolis, B. and Pizzutilo, S.: 1994, 'User-Tailored Hypermedia Explanations'. Proceedings of the Workshop on Adaptive Hypertext and Hypermedia at the Fourth International Conference on User Modelling.
(URL: http://www.cs.bgsu.edu/hypertext/adaptive/deRosis.html).

Eklund, J.: 1995, 'Cognitive Models for Structuring Hypermedia and Implications for Learning from the World Wide Web'. Proceedings of the First Australian World Wide Web Conference, pp. 111-116.

Eklund, J. and Zeiliger, R.: 1996, 'Navigating the Web: Possibilities and Practicalities for Adaptive Navigational Support'. Proceedings of the Second Australian World Wide Web Conference, **page number? URL?**.

Fink, J.; Kobsa, A.; Nill, A. (1996): User-oriented Adaptivity and Adaptability in the
AVANTI Project. Conference "Designing for the Web: Empirical Studies", Microsoft Inc., Redmond, WA.
(URL: http://zeus.gmd.de/hci/projects/avanti/publications/ms96.html

Hammond, N. 1993, 'Learning with Hypertext: Problems, Principles and Prospects'. In: C. McKnight, A. Dillon, and J. Richardson (eds.): *Hypertext - A Psychological Perspective,* Ellis Horwood, pp. 51-69.

Hirashima, T., Hachiya, K., Kashihara, A. and Toyoda, J.: 1997, 'Information Filtering Using User's Context on Browsing in Hypertext'. *User Modeling and User-Adapted Interaction* (this issue).

Hubler, A.W. and Assad, A.M.: 1995, 'CyberProf, an Intelligent Human-Computer Interface for Asynchronous Wide Area Training and Teaching'. Proceedings of the Fourth World Wide Web Conference, pp. 231-238.

Johnson-Laird, P.N.: 1983, *Mental Models*. Cambridge University Press.

Jonassen, D.: 1991, 'Objectivism Versus Constructivism: Do We Need A New Philosophical Paradigm?'. *Educational Technology, Research and Development* **39**(3), pp. 5-14.

Kahneman, D.: 1973, *Attention and Effort*. Prentice-Hall, Englewood Cliffs, NJ.

Kay, J. and Kummerfeld, B.: 1994a, 'An Individualised Course for the C Programming Language'. Proceedings of the Second International WWW Conference.
(URL:
http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Educ/kummerfeld/kummerfeld.html).

Kay, J. and Kummerfeld, B.: 1994b, 'Adaptive Hypertext for Individualised Instruction'. Proceedings of the Workshop on Adaptive Hypertext and Hypermedia at the Fourth International Conference on User Modelling.
(URL: http://www.cs.bgsu.edu/hypertext/adaptive/Kay.html).

Linard, M.: 1993, 'For a Knowing Subject in Models of Learning'. *Communication and Cognition* **26**(3/4), pp. 291-320.

Linard, M. and Zeiliger, R.: 1995, 'Designing an Educational Support for an Educational Software'. Proceedings of the 5th East-West International Conference on Human-Computer Interaction, pp. 130-145.

Marshall, C.C. and Irish, P.M.: 1989, 'Guided Tours and On-line Presentations: How Authors Make Existing Hypertext Intelligible for Readers'. Proceedings of Hypertext '89, pp. 15-26.

Messick, S.: 1976, *Individuality in Learning.* Jossey-Bass, San Francisco.

Miller, A.: 1991, 'Personality Types, Learning Styles and Educational Goals'. *Educational Psychology* **11**(3/4), pp. 217-238.

Pérez, T. J. Gutiérrez, and P. Lopistéguy: 1995a, 'An Adaptive Hypermedia System'. AI-ED'95, 7th World Conference on Artificial Intelligence in Education, Washington, DC, pp. 351-358.

Pérez, T., P. Lopistéguy, J. Gutiérrez and I. Usandizaga: 1995b, 'HyperTutor: From Hypermedia to Intelligent Adaptive Hypermedia'. ED-MEDIA'95, World Conference on Educational Multimedia and Hypermedia, Graz, Austria, pp. 529-534.

Reed, W.M., Oughton, J.M., Ayersman, D.J., Giessler, S.F. and Ervin, J.R.: 1995, 'Computer Experience and Learning Styles: Linear Versus Non-Linear Navigation'. Proceedings of ED-MEDIA'95, p. 802.

Riding, R.J. and Rayner, S.: 1995, 'The Information Superhighway and Individualised Learning'. *Educational Psychology* **15**(4), pp. 365-378.

Steinberg, E. R.: 1989, 'Cognition and Learner Control: A Literature Review 1977-1988'. *Journal of Computer Based Instruction* **4**, 117-121.

Thüring, M., Haake, J. and Hannemann, J.: 1991, 'What's ELIZA Doing in the Chinese Room? Incoherent Hyperdocuments - And How to Avoid Them'. Proceedings of Hypertext '91, pp. 161-177.

Tomek, I., Maurer, H. and Nassar, M.: 1993, 'Optimal Presentation of Links in Large Hypermedia Systems'. Proceedings of ED-MEDIA'93, pp. 511-518.

van Dijk, T.A. and Kintsch, W.: 1983, *Strategies of Discourse Comprehension.* Academic Press, Orlando.

Van Ginderen, B.: 1990, 'An Object-Oriented Hypertext System for Computer-Aided Learning'. Masters Thesis, Eindhoven University of Technology.

Weber, G. and Specht, M.: 1997, 'User modeling and adaptive navigation support in WWW-based tutoring systems'. In: A. Jameson, C. Paris and C. Tasso (eds): User Modeling: Proceedings of the Sixth International Conference UM97. Springer-

Verlag, Wien New York, pp. 289-300. (URL: http:// http://um.org/um_97/gz/WeberG.ps.gz)

Zeiliger, R.: 1993, 'Adaptive Testing: Contribution to the SHIVA Model'. In: D. Leclercq and J. Bruno (eds.): *Item Banking: Interactive Testing and Self-Assessment*, NATO ASI Serie F, **112**, Springer-Verlag, Berlin, pp. 54-65.

Authors' Vitae

Licia Calvi
University of Antwerp (UIA), Department of Romance Languages and Literatures, Universiteitsplein, 1, 2610 Antwerp, Belgium.

Licia Calvi ia a Ph.D. candidate in Romance Languages at the University of Antwerp. She received her B.A. in Philosophy from the University of Pavia (Italy) in 1991, and her M.S. degree in Artificial Intelligence and Cognitive Science from the University of Leuven (Belgium) in 1993. Her primary interests lie in the areas of cognitive aspects of human-computer interactions, adaptive hypermedia and intelligent tutoring systems. Her paper summarizes the current state of her thesis work on adaptive hypermedia. Other research interests include computer-assisted language learning.

Paul De Bra
Eindhoven University of Technology, Department of Computer Science, Postbus 513, 5600 MB Eindhoven, The Netherlands.

Paul De Bra is Professor of Computer Science at the Eindhoven University of Technology. He received his B.A. in Mathematics and Computer Science from the University of Antwerp in 1981 and his Ph.D. in Computer Science from the same university in 1987. Since September 1994, he is head of the Information Systems group in the Computer Science Department. His primary interests lie in the areas of hypermedia systems and object-oriented databases. Currently he is heavily involved in the development of the World Wide Web, dealing with issues such as information retrieval, browsing semantics, navigation aids, and collaborative authoring.