2L670: A Flexible Adaptive Hypertext Courseware System

Paul De Bra[1]
Eindhoven University of Technology
Department of Computing Science
PO Box 513
NL 5600 MB Eindhoven, The Netherlands
Email: debra@win.tue.nl

Licia Calvi
University of Antwerp (UIA)
Department of Romanic Languages
Universiteitsplein 1
B 2610 Wilrijk, Belgium
Email: calvi@uia.ua.ac.be

ABSTRACT
In [4, 5] (among other papers) we have reported on the development of an adaptive hypertext document and system, used for learning about the subject of hypertext, through distance learning by means of World Wide Web. In the terminology of Brusilovsky's overview paper [1], the system offered adaptive content and link hiding. This short paper briefly describes the latest developments, which include the possibility for users to choose between link hiding and link annotation. The adaptive hypertext contents consists of standard HTML (3.2) pages, which makes it easy for authors to create adaptive courses using off the shelf authoring tools.

KEYWORDS: hypertext courseware, adaptive content, adaptive hiding and annotation

INTRODUCTION
Adaptive hypertext and hypermedia is an emerging research field, on the crossroad between hypermedia and intelligent user-model-based adaptive systems. Although there are many application areas for adaptive hypermedia systems, the vast majority of these systems is used for educational purposes, as intelligent tutoring systems. Based on a user-model the system generates (lists of) links to nodes (pages) the student should study in order to achieve a desired learning goal.

The ELM-ART adaptive Lisp course [2] and the subsequently developed general-purpose adaptive hypertext system Interbook [3] are typical for adaptive hypertext: the dynamically determined link structure is presented through *frames* (in HTML terms). Interbook presents a table-of-contents-like frame in which links are annotated to indicate which pages are suggested, advised against, or neutral. Information pages are presented in another frame. The hypertext links that occur in these pages are also annotated, but the information content is not adaptive.

This paper reports on the development of the adaptive courseware 2L670 (named so for historic reasons), which features adaptive content as well as adaptive linking. Authoring for this courseware is not only simple but also familiar because it uses standard HTML.

ADAPTIVE CONTENT
Depending on the user's knowledge state information on a given subject may need to be presented in different ways. Students who are first reading about hypertext may be confused when they see the term "node" whereas the word "page", used in the same context, would be meaningful to them, and probably sufficiently accurate in an introductory text. In the course text for "2L670: Hypermedia Structures and Systems", the use of adaptive hypertext requires that students first visit a "readme" page with instructions on how to use the courseware and how to configure their WWW-browser. Therefore a short paragraph which tells students to go to the readme page is prominently displayed, along with a link to that page. After reading the instructions the textual content of the start-page of the course is changed automatically. The pointer to the readme page is removed from the top, and a small reminder at the bottom of the page is all that remains.

Such adaptive changes to a page, using what Brusilovsky [1] calls *fragment variants*, are non-trivial to realize in World Wide Web, because HTML does not allow for text fragments which are *condionally* made visible or hidden. It would be possible to use *Dynamic HTML* for this purpose, but this would lead to HTML documents that are complicated (because of the JavaScript or VBscript code they need to contain) and thus difficult to author.

The (latest) courseware for 2L670 implements adaptive content in HTML by means of a preprocessor that filters content fragments by means of conditionals encoded in structured HTML comments:

```
<!-- if definition and history -->
 This part appears if the two "concepts"
 definition and history are both known
 according to the user model.
<!-- else -->
 If this is not the case then this
 alternative is presented instead.
<!-- endif -->
```

Because the comments have no "meaning" in HTML, there is no need to respect the proper nesting of conditionals and HTML tags. We give an example of the (useful) use of non-properly nested conditionals and HTML tags (in the next section).

[1]Paul De Bra is also affiliated with the University of Antwerp and the "Centrum voor Wiskunde en Informatica" in Amsterdam.

ADAPTIVE LINKING

Brusilovsky [1] distinguishes five types of adaptive linking: *direct guidance*, *adaptive link sorting*, *adaptive link annotation*, *adaptive link hiding* and *map adaptation*. Most adaptive hypertext systems offer only one of these features. Interbook [3] for instance concentrates on *link annotation*, meaning that desired links are marked differently from undesired links, in this case by means of a colored (big) dot.

The (latest) courseware for 2L670 supports two variations of adaptive hiding as well as adaptive navigation.

- Adaptive link annotation in 2L670 is realized by means of link classes which are turned into different colors by using *cascading style sheets*. Because of differences in author and user preferences the courseware can be configured to use any color scheme (for desired, undesired, neutral and external links) the author desires, and the user may override these choices through a "setup" HTML page and a CGI-script. Links that may be desired or undesired at times are marked in the HTML source of the pages using a link class we call "conditional". The 2L670 preprocessor translates this to the classes "good" or "bad" depending on the user model.
- One variation of adaptive link hiding is to make the link anchors indistinguishable from normal text. Undesired links are still present, but the user is not attracted to them because they look like plain text, whereas the desired (and neutral) links are clearly visible. The default setup of the courseware uses *blue* for desired links, *purple* for neutral links (which indicate that the links lead to pages with no new information because the user already visited them) and *black* for undesired links. Both the author and the user may configure the courseware to use link annotation or this variation of link hiding simply by replacing *black* by another color if desired.
- The second variation of adaptive link hiding, also called link removal, means that a link is made non-functional when not desired. This variation can be achieved through adaptive content, as in the following example:

```
<!-- if desired -->
 <a href="...">
<!-- endif -->
 here is the link anchor text
<!-- if desired -->
 </a>
<!-- endif -->
```

Note that if the conditionals were not HTML comments, but "real" tags, this construction would violate the nesting rules.

AUTHORING FOR 2L670

2L670 is different from other educational adaptive hypertext courseware in that it combines adaptiveness with "normal" hypertext functionality. A hyperdocument for 2L670 consists of HTML nodes and links. All features of HTML can be used by the author, including frames, scripting (with JavaScript or VBscript), Java applets, etc. In addition to hypertext functionality some links can be annotated (or hidden) conditionally by means of the "conditional" link class, and text fragments (as small as a single word and as large as an entire page) can be conditionally included through the `<!-- if -->` comments. This approach has the advantage (over other adaptive hypertext systems for the Web) that authoring can be done using standard HTML editors or word processors capable of generating HTML (with comments).

Unfortunately, the 2L670 courseware does not eliminate the need for authors to carefully construct a didactic plan which indicates how *concepts* depend on each other. Each HTML page in a course starts with two structured comments:

```
<!-- requires definition and history -->
<!-- generates xanadu -->
```

for instance means that links to this page are only desired when the user has learnt about the concepts "definition" and "history", and that after reading this page the user has learnt the concept "xanadu". The expresion in the *requires* clause can be any boolean expression using concept names (and possibly including subexpressions in parentheses). These two types of clauses form the basis through which an author establishes the dependencies between concepts. The system automatically annotates or hides links to guide the user towards pages in an order which corresponds to the order the author has envisaged when creating the dependencies between concepts.

1 CURRENT AND FUTURE DEVELOPMENTS

The 2L670 courseware is currently being used for the course "Hypermedia Structures and Systems" (initially with the code 2L670) which is offered to students at six universities in The Netherlands and Belgium. It is also used in a course on Graphical User-Interfaces. The software for the courseware is freely available upon demand. It consists of two (Fast-)CGI scripts written in Java and a few wrapper shell scripts. Apart from adaptive hypertext distance education requires additional support, like a message board (for which we use HyperNews), multiple-choice tests (for which the 2L670 courseware also offers support), and upload facilities for assignment work (for which we plan to use the DReSS document repository system).

REFERENCES

1. Brusilovsky, P., Methods and Techniques of Adaptive Hypermedia, User Modeling and User-Adapted Interaction 6, Kluwer academic publishers, pp. 87–129, 1996.
2. Brusilovsky, P., Schwarz, E., Weber, G., ELM-ART: An intelligent tutoring system on World Wide Web, Proc. of the Third International Conference on Intelligent Tutoring Systems, ITS-96, Montreal, 1996. (Lecture Notes in Computing Science, vol. 1086, pp. 261–269).
3. Brusilovsky, P., Schwarz, E., Weber, G., A Tool for Developing Adaptive Electronic Textbooks on WWW, Proc. of the WebNet'96 Conference, pp. 64–69, San Francisco, 1996.
4. Calvi, L., De Bra, P., Improving the Usability of Hypertext Courseware through Adaptive Linking, Proc. of the ACM Conference on Hypertext, pp. 224–225, Southampton, 1997.
5. De Bra, P., Calvi, L., Creating Adaptive Hyperdocuments for and on the Web, Proc. of the AACE WebNet'97 Conference, pp. 149–154, Toronto, 1997.