Multimedia Adaptation using AHA!

Paul De Bra, Natalia Stash
Eindhoven University of Technology (TU/e)
Department of Computer Science
Eindhoven, The Netherlands
{debra, nstach}@win.tue.nl

Abstract: Presentation formats used for publishing information on the World Wide Web are constantly evolving. Simultaneously, the range of devices used to access this information is expanding. In education adaptive hypermedia methods and techniques are often used to tailor a presentation to the individual user. Typical adaptation software can only adapt simple presentation formats like HTML (with embedded objects that are not adapted). The evolution in Web technology requires adaptive hypermedia software to keep up, and to provide adaptation for documents in new formats such as SMIL and (X)HTML+SMIL, and to provide adaptation to more than just the user's knowledge level and learning style but also the properties of the user-interface device and the network connection. This paper shows how the AHA! system can be used to provide adaptive interaction with SMIL documents, and in fact with almost any XML-based format. Adaptation in SMIL and (X)HTML+SMIL using AHA! is currently being used in an assignment for an adaptive hypermedia course being taught at the Eindhoven University of Technology. Demo's from this assignment are available from the AHA! project website (http://aha.win.tue.nl/).

Introduction

Many adaptive educational hypermedia applications (and systems) exist to date. Often the term *adaptive hypermedia* is an overstatement for what these systems actually offer: *link annotation or hiding* and/or *canned text adaptation* in the terminology of Brusilovsky's overview paper (Brusilovsky, 2001). Most of the evolution in the field of adaptive educational applications is in *what* the application adapts *to*. But simultaneously we see some evolution in *what* can be adapted in the systems as well. There are too many systems to name all of them, but we mention a few to illustrate the evolution in the field. Systems like *InterBook* (Brusilovsky et al, 1998), *AdaptWeb* (Freitas et al, 2002) and others provide link annotation and the generation of some overviews based on the user's knowledge of different concepts. The author has to prepare an Electronic Textbook as a specially structured file or set of files (MS Word for *Interbook*, XML for *AdaptWeb*) which is or are later presented to the user in HTML format. *AdaptWeb* is adding a "Content Ontology" and adaptation to the user's learning style. *iWeaver* (Wolf, 2002) is another interactive web-based adaptive learning environment that aims to provide adaptation to specific learning styles. This adaptation is mostly done through media selection, using text, audio and PowerPoint slides. *iWeaver* uses SMIL to perform the synchronization of audio with slide shows. In *iWeaver* there is no link adaptation. In some systems the pages are not just *adapted* but *generated* dynamically during the interaction between the user and the system. In *Tangow* (Carro et al., 1999), for example, each page is composed "on the fly" from general information about the type of media elements associated to each task and their layout. The concrete media elements that appear in these pages are selected depending on learners' characteristics such as age and language.

AHA! started out as a system that could adapt the contents and the link anchors based on the user's knowledge, and this adaptation was done using a extension of HTML. AHA! version 2.0 added adaptation to arbitrary aspects of the user's relation with the application domain (using a rich and flexible user model capable of handling arbitrary concepts and attributes) and the adaptation was done using an extension of XHTML. AHA! 3.0 has gone one step further and can perform adaptation to documents in any XML format. When a user requests a resource from an AHA! server the resulting content may vary depending on the current state of the user model. Fragments or objects can be conditionally included or excluded. In this way the system provides *adaptive presentation* support. The user can be guided towards "desirable pages", corresponding to his knowledge level, preferences and other characteristics, and away from "undesirable pages". This is done through *link annotation* and *link hiding*). This results in *adaptive navigation support*. AHA! only requires that the used XML format has a notion of "link" (like the <a> tag in HTML) and of "included object" (like the <object> tag in HTML).

Most adaptive hypermedia systems are good for a specific type of application and a specific type of presentation. In previous work we have already shown that AHA! is suitable for different application areas (De Bra et al, 2002a, 2002b). In this paper we stress that AHA! is also suitable for different presentation types. AHA! can provide adaptation for different XML formats. We illustrate how this contributes to the ability to perform multimedia presentation adaptation by means of adaptation in SMIL and in (X)HTML+SMIL format. (For in-depth information on SMIL and (X)HTML+SMIL we refer to www.w3.org/AudioVideo/.)

Content Adaptation in AHA!

When a hypermedia application offers many ways to navigate to the same information page users will not have the same knowledge when they reach that page. Differences between users can be taken into account when deciding which *additional* or *prerequisite explanations* to include. Also, different users may have different media preferences. Some users prefer textual explanations and others prefer images, audio or video. AHA! supports the *conditional inclusion of fragments* to provide this content adaptation. This technique is offered in slightly different ways in the three page formats that AHA! supports:

- The oldest page format is an extension of HTML. Since HTML is still the most popular markup language on the Web, and since AHA! allows an application to link to and perform adaptation on external documents (residing on other servers but served "through" AHA!), the HTML format is still supported. In AHA!-extended HTML an <if> tag can be used to conditionally include a fragment of text. An example:

  ```
  <if expr="c2L690.xanadu.knowledge &lt; 50">
    <block>
      Include additional explanation
    </block>
  </if>
  ```

  An additional explanation is added when the "knowledge" of the concept "xanadu" in the course "c2L690" is less than 50. Note that we must write < to prevent the < sign from being interpreted as the start of a tag.
- In AHA! 2.0 (De Bra et al, 2002a) the use of XHTML was introduced, and an extension defined to allow the use of <if> tags, just like with the HTML format. The XHTML+AHA! format is defined using XHTML modularization. The AHA! engine looks at the AHA! tags to perform the conditional inclusion, and produced a document in standard XHTML format. AHA! 2.0 is actively being used by several research teams in different countries. It is also serving the hypermedia course 2L690 at our university.
- In AHA! 3.0 (De Bra et al, 2003) an alternative way to perform the conditional inclusion of fragments was introduced, by using the <object> tag. This tag exists in XHTML to (unconditionally) include an object stored in a separate file. The AHA! engine treats object tags with a special "aha/text" type as conditional. The engine uses rules that are defined in the concept structure to decide which actual object (also called "base object") to include. As a result conditional fragment inclusion is realized without the need for extensions to the XHTML language. Furthermore, since nothing in AHA! still depends on the XHTML language, other XML formats can be parsed and served as well. The only requirement for AHA! to perform content adaptation is the availability of an <object> tag, or another tag that can be used for the same purpose (and that must be defined as such in the AHA! software). Below is an example of the conditional inclusion of an object:

  ```
  <object name="ConditionalObject" type="aha/text" />
  ```

  In figure 1 we show a situation where the user working with an AHA! application requests a Web-page which consists of some static text and an included object. When the page is accessed one of the base-objects is included into the resulting page on the basis of information in the user model. The selected base-object is inserted in the stream that AHA! parses. As a consequence a base-object can conditionally include other objects recursively. A badly designed application could have objects that include themselves (directly or indirectly) in an infinite recursive loop. AHA! limits the number of object inclusions to abort such a loop.

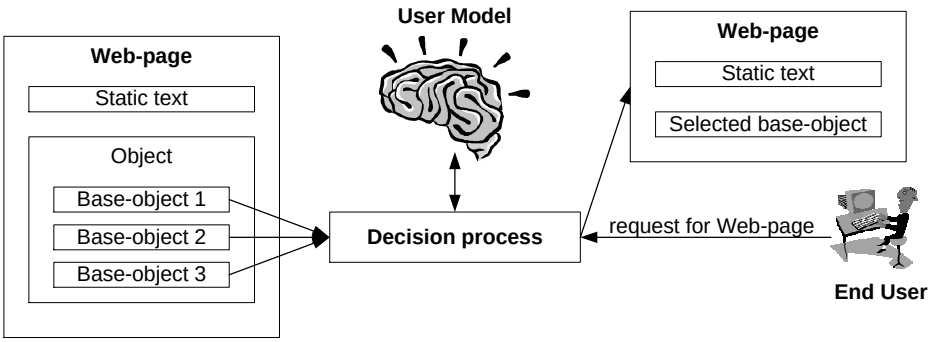

Figure 1. Conditional inclusion of objects in AHA! pages.

Apart from enabling the use of standard XHTML or other XML formats the object inclusion also has the advantage that the adaptation rules are all defined in one place (the concept list) instead of being scattered over and embedded in the XHTML pages.

Link Adaptation in AHA!

AHA! performs *link annotation* and/or *link hiding* by means of style sheets. Links that must be adapted are defined as being of the *class* "conditional".

```
<a href="destination.xhtml" class="conditional">this is an adaptive link</a>
```

The AHA! engine replaces the word "conditional" by "good", "neutral" or "bad". This results in different link colors, provided that the page includes (or refers to) the appropriate stylesheet that defines the presentation for these link classes. Existing AHA! applications (like our on-line hypermedia course 2L690) often define these colors as blue, purple and black, which results in *link hiding*. But when using different colors (e.g. green, orange, red) the link adaptation becomes *link annotation*.

Apart from HTML and XHTML, link adaptation can be performed to documents in other XML formats that have an anchor tag (called <a>) and that allow for the inclusion of stylesheets.

Adaptation in SMIL

The possibilities of HTML (and XHTML) for creating true multimedia applications are limited. W3C has therefore developed a new multimedia markup language, SMIL, for *Synchronized Multimedia Integration Language*. SMIL is a markup language vaguely similar to (X)HTML (and defined as an XML language) that combines powerful multimedia and timing capabilities with basic layout and formatting. To some extent, SMIL plays the same role in a SMIL player that HTML plays in a Web browser (namely providing information about how to layout and format a page). SMIL pages must be displayed using special SMIL players. RealOne is the most popular (free) SMIL 2.0 player (see www.real.com), but the Open Source Ambulant player developed at CWI (see ambulant.sourgeforce.net or www.cwi.nl/projects/Ambulant/distPlayer.html) is becoming an interesting alternative).

A SMIL presentation can consist of multiple components of different media types (such as video, audio, text, and graphics), linked via a synchronized timeline. For example, in a slide show, the corresponding slide can be displayed when the narrator in the audio starts talking about it. This is done in the adaptive *iWeaver* for instance (Wolf, 2002). SMIL files can have links, using the <a> tag, to other SMIL files that are shown in the same window or in a different window.

The SMIL language itself contains an "adaptation" or "alternate content" mechanism. Using the <switch> tag and "test attributes" one can have a SMIL player choose between alternative content. Examples of attributes the player can use are "systemBitrate" to select content that fits the current network bandwidth, "systemCaptions" to choose between video with and video without captions, "systemLanguage" to select content in a given language, "systemScreenDepth", "systemScreenSize", etc. These adaptation features enable a SMIL player to adapt to technical circumstances and some fairly static user preferences. They do not allow for adaptation to a user's

evolving knowledge level or to cognitive style aspects that influence the order of a presentation or the selection of items in different media (for other than technical reasons).

Recently within the AHA! project we started research into the possibility of providing adaptation to various learning styles. There are many classifications of cognitive or learning style and even some conflicting definitions of what the proposed terms mean. We do not want to enter into that discussion, but focus on how to perform adaptation, using AHA!, to some of the commonly recognized learning styles. Kolb (Kolb, 1984) defines learning styles as the "various methods individuals have for perceiving and processing information while reacting to their environment". Two "methods" that we wish to support are the selection of media items (and types) to accommodate for *visualizers* who prefer images and video and *verbalizers* who prefer text and possibly spoken audio. Another type of adaptation relates to the order in which different information items are processed. Some learners prefer to learn about a general principle by first studying some examples and others prefer to first study the principle and then look at examples. Some learners prefer to first get a broad overview of a field to study and then go into details whereas others prefer to take the first topic and study the details of that before moving to the next topic. Both kinds of adaptation (media selection and presentation order) can be performed in AHA! through AHA!'s ability to perform conditional object inclusion. It can also be supported by the link adaptation, but because adaptive sequencing (through AHA!) is already possible within a single SMIL file we will concentrate on content adaptation. We briefly show some of the *adaptation rules* that are needed to perform the object selection, but focus on how to refer to conditional content in the SMIL files.

In AHA! a domain model concept can be represented by different types of media if we define it as an object concept. In XHTML an <object> tag is used to indicate where a conditional object should be included. In SMIL we use the <ref> tag instead.

A SMIL presentation may include the following tag:

```
<ref  src="ConditionalObject" type="aha/text" />
```

An XHTML presentation would include:

```
<object  src="ConditionalObject" type="aha/text" />
```

Apart from "ref" versus "object" there is no difference. The type "aha/text" just means that the AHA! engine will process the tag, look for the "ConditionalObject" concept and select the appropriate *resource* to include in the SMIL file instead of the given <ref> tag. Typically the *resource* will be a one line SMIL fragment with a reference to a media item but it may also be a file with a larger SMIL fragment, as long as it is syntactically correct. The type "aha/text" does not mean the object is text; it is just a type to indicate that the AHA! engine should consider it for conditional inclusion of objects. Below is an example of a SMIL file containing conditional objects:

```
1     <smil>
2       <head>
3         <layout>
4           <region id="main" top="10" left="20" height="200" width="200" z-index="3"> </region>
5         </layout>
6       <head>
7       <body>
8         <seq repeat="1">
9           <ref dur="10s" fill="remove" repeat="1" src="textfile.html" />
10          <ref src="application.smilobject" type="aha/text" />
11        </seq>
12      </body>
13    </smil>
```

Lines 9 and 10 specify fragments that are shown. The first fragment (or any other type of media) is displayed for 10 seconds and then disappears. The first fragment, "textfile.html" need not really be an HTML file. The SMIL player should figure out what the real type is when loading the file. Line 10 presents an adaptively included object where "application" is the name of the application and "smilobject" is the name of a concept of that application. When AHA! processes the SMIL page it executes adaptation rules associated with the concept "application.smilobject" in order to decide which piece of SMIL code to put in place of the "ref" tag on line 10.

Assume there are 3 resources associated with "application.smilobject" and each of them represents this object under different conditions. These conditions are expressions which can use information about the user from the user profile, the values of the attributes of the domain model concepts as well as some user related information which is stored in the pseudo-concept *personal* (De Bra et al., 2002a). Assume the author of an application wants to

take into account the user's preferences for certain types of media. For this purpose he can define a "MediaTypePreference" attribute of the "personal" concept. (In AHA! an author can create arbitrarily many concepts with arbitrarily many attributes.) The value of this attribute can be initialized through a registration form in which the user is asked to choose between various media types. The author can specify the object's behavior in the following way:
if personal.MediaTypePreference == "audio" then represent "application.smilobject" with *audio_file.smil*
if personal.MediaTypePreference=="video" then represent "application.smilobject" with *video_file.smil*
else (if none of the previous conditions is satisfied) represent "application.smilobject" with *default_file.smil*.
AHA! offers two authoring tools, the low-level Concept Editor and high-level Graph Author (Stash & De Bra, 2003) to define these rules. The if-then-else rules above do not show the real syntax used by AHA! but are more readable.

The files *audio_file.smil*, *video_file.smil* and *default_file.smil* should be well-formed documents in order for AHA! to be able to parse them, and check for possible recursively included objects (De Bra et al, 2003). The resource pages may be defined in the following way (let's take the *video_file.smil* which shows a video clip to the user):

```
<video dur="10s" fill="remove" repeat="1" src="video_file.mpg" />
```

When the resource contains several tags a root element must be added. For instance, when the adaptation wishes to adapt to a preference for examples first or definitions first the alternative resource pages may be:

```
<smil>
  <video dur="30s" fill="remove" repeat="1" src="video_example1.mpg" />
  <video dur="30s" fill="remove" repeat="1" src="video_example2.mpg" />
  <ref src="textual_definition.html" type="text/html">
</smil>
```

For a "definitions first" adaptation a resource with different order for these fragments would be used instead. Note that because this fragment contains more than one tag it needs an additional root element in order to be well-formed.

The adaptation that is possible in AHA! allows one "ref" tag to be replaced by small or large SMIL fragments, containing just one or many media objects, shown in parallel, in sequence or any other combination. It also allows the included resource to contain more "aha/text" <ref> tags, resulting in recursive object inclusion. It is very important to carefully consider the choices of media items and presentation order (Wolf, 2002). Mayer's experiments (Mayer, 2001) show that particular combinations of media are advantageous for learning, whereas other combinations have a detrimental effect, either for some or for all types of users. The use of too many media items was also found to be detrimental in this research. Mayer described this as the "redundancy principle".

Adaptation in (X)HTML+SMIL

Using AHA! it is possible to influence the layout of a SMIL application by including layout information within the objects' tags. SMIL allows this by using stylesheet information. However, in SMIL the layout definitions really belong in the header, where <ref> tags are not allowed. Microsoft's alternative to a complete SMIL implementation, first using HTML and later XHTML plus part of SMIL makes it possible to perform adaptation to the layout as well as the content because in (X)HTML layout information is part of the body of the document, where <object> tags are allowed. The <object> tags are used by AHA! to conditionally insert pieces of (X)HTML, which can contain content as well as layout information.

W3C has adopted Microsoft's idea and published the XHTML+SMIL profile. Essentially, XHTML+SMIL extends XHTML with the SMIL modules that deal with anything but the layout. Using XHTML+SMIL it is relatively easy to create presentations in which text, animations, videos and other media items are combined and presented in different parts of a page, appropriately synchronized. Using AHA! it is possible to adapt the content of such applications, using the "object" tag for conditionally included objects or using the "if" tag that exists in the XHTML+AHA! format.

At our university we recently started a new course on the topic of adaptive hypermedia. For this course there are viewgraphs, text pages and video fragments (one fragment for each viewgraph). To make the course itself adaptive students will use these new standards to create an (X)HTML version of the course material as well as of some of their own multimedia material. The adaptive features of AHA! can be used more easily in (X)HTML+SMIL than in pure SMIL because it is easier to perform layout adaptation and adaptive link annotation when the basis of the pages is (X)HTML.

We show a short fragment of HTML+TIME as defined by Microsoft, combined with a fragment that is conditionally included using the AHA! <object> tag. This example shows timing but cannot include audio or video in this way.

```
1       <html>
2         <head>
3           <style>
4                   .time    {behavior: url(#default#time2);}
5           </style>
6         </head>
7         <body>
8           <p>This text appears right away. More lines to follow...</p>
9           <p class="time" begin="2" dur="5" >This appears after 2 seconds.</p>
10          <p class="time" begin="4" dur="5">This appears after 4 seconds.</p>
11          <p>This is the last line.</p>
12          <!-- The following object is conditionally included or replaced -->
13          <object data="application.htmltimeobject" type="aha/text" />
14          <a href="page.html" class="conditional">here is a link that AHA! will adapt</a>
15        </body>
16      </html>
```

The HTML+TIME example uses the Cascading Style Sheets (CSS) class attribute (lines 3-5) to indicate that timing is used. The AHA! engine only looks at "object" and "a" tags (lines 13 and 14), and as a result it ignores the timing information. Hence, adding Microsoft's timing implementation in HTML files does not influence AHA!'s ability to perform adaptation to these files. For this example to work the object that is conditionally included based on the rules for the "htmltimeobject" concept of "application" must be syntactically correct. AHA!'s HTMLHandler class requires this. An example could be:

```
<html>
  <head></head>
  <body>
    <img class="time" begin="8" dur="5" src="image.jpg" />
  </body>
</html>
```

As the example shows, the included fragment may use timing controls just like the main HTML file.

To include media items into the presentation with elements as *t:MEDIA*, *t:AUDIO* and *t:VIDEO* the HTML+SMIL definition requires that the html tag declare the XML namespace *t:* as shown in the following code: code:

```
<html xmlns:t="urn:schemas-microsoft-com:time">
```

To establish *t:* as the namespace the *time2* behavior is imported into the namespace:

```
<?IMPORT namespace="t" implementation="#default#time2">
```

AHA!'s HTMLHandler cannot parse the documents which use XML namespaces. The solution is to use XHTML+SMIL. The following example works with AHA! (because AHA! uses its XMLHandler for it instead of its HTMLHandler):

```
1       <!DOCTYPE html SYSTEM "../../www.w3c.org/DTD/xhtml1-strict.dtd">
2       <html xmlns="http://www.w3.org/1999/xhtml"
3               xmlns:t="urn:schemas-microsoft-com:time">
4         <head>
5           <?IMPORT namespace="t" implementation="#default#time2"?>
6         </head>
7         <body>
8           <object name="xhtmltime.object" type="aha/text" />
9           <!-- The following object is conditionally included or replaced -->
10          <a href="page.html" class="conditional">here is a link that AHA! will adapt</a>
11          …
12        </body>
13      </html>
```

If the resource page conditionally associated with the “xhtmltime.object” concept includes a media element then the same namespace definition is also needed in the fragment:

```
<html xmlns="http://www.w3.org/1999/xhtml"
        xmlns:t="urn:schemas-microsoft-com:time">
  <head>
    <?IMPORT namespace="t" implementation="#default#time2"?>
  </head>
  <body>
    <t:audio src=“audio_file.au” />
  </body>
</html>
```

While this looks complicated (9 lines to just include one audio fragment) it is very uniform and can thus easily be created using a standard template.

Conclusion

In this paper we addressed the problem of combining adaptive educational hypermedia with real multimedia presentation. SMIL and (X)HTML+SMIL already allow some kind of adaptation, but that is mostly to technical aspects, not the user’s knowledge about the concepts of a course. AHA! provides a platform for combining SMIL with adaptation to the user. It can be used with “pure” SMIL, with Microsoft’s HTML+SMIL, or actually HTML plus the timing aspects of SMIL, and with XHTML+SMIL which is XHTML plus everything of SMIL except the layout part.

In figure 2 we present the general picture of how AHA! works with the documents of different formats. The pages written in HTML+AHA!, XHTML+AHA! and (X)HTML+SMIL formats are translated by the AHA! engine into XHTML pages and sent to the browser. SMIL pages are displayed with a SMIL player. The tags that can be used by AHA! for adaptation are currently fixed. Link adaptation requires an <a> tag. Content adaptation requires an <object> or <ref> tag. The required changes in the AHA! software to support XML formats that use different tags to indicate links and objects are minimal (but not zero).

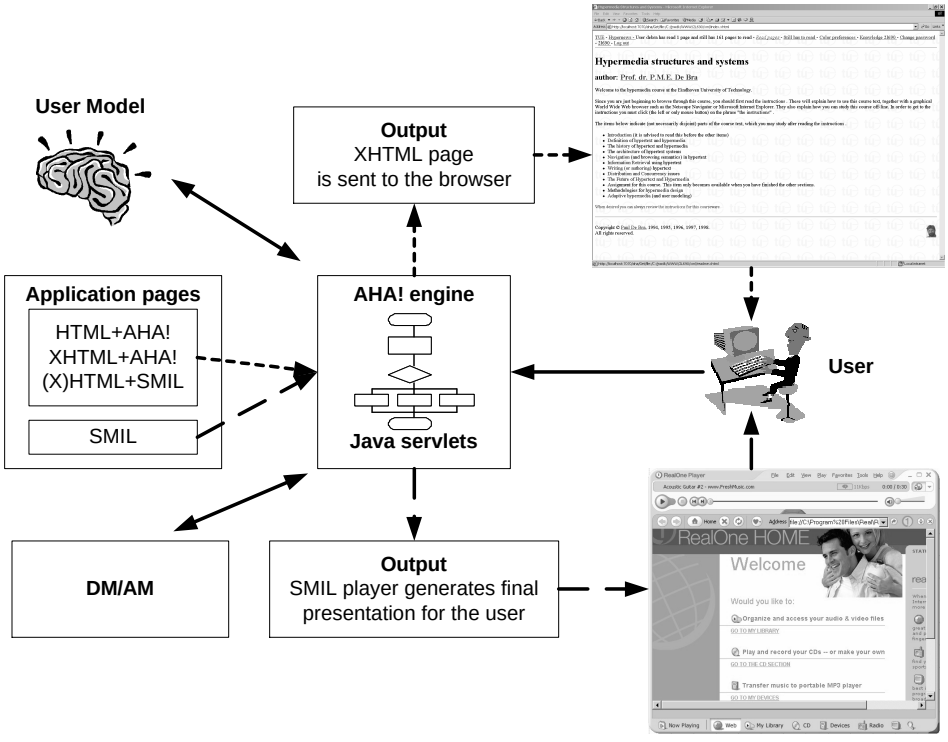

Figure 2. Processing documents with AHA!.

Acknowledgement

This work is supported by the NLnet Foundation and by the Minerva ADAPT project (101144-CP-1-2002-NL-MINERVA-MPP). It will also be further developed and communicated thanks to the EU FP6 Network of Excellence PROLEARN.

References

Brusilovsky, P. (2001). Adaptive hypermedia. *User Modeling and User Adapted Interaction, Ten Year Anniversary Issue, 11* (1/2).

Brusilovsky, P., Eklund, J., and Schwarz, E. (1998). Web-based education for all: A tool for developing adaptive courseware. *Computer Networks and ISDN Systems* (Proceedings of Seventh International World Wide Web Conference, 14-18 April 1998) 30 (1-7), 291-300.

Carro, R. M., Pulido, E., & Rodríguez, P. (1999). Task-based Adaptive learNer Guidance On the WWW, *Second Workshop on Adaptive Systems and User Modeling on the World Wide Web*, 49-57, (available online at http://wwwis.win.tue.nl/asum99/carro/carro.html.

De Bra, P., Aerts, A., Smits, D., Stash, N. (2002a). AHA! Version 2.0, More Adaptation Flexibility for Authors. In *Proceedings of the AACE ELearn'2002 conference*, October 2002, 240-246.

De Bra, P., Aerts, A., Smits, D., Stash, N., (2002b). AHA! meets AHAM. *Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, May 2002. Springer LNCS 2347, 381-384.

De Bra, P., Aerts, A., Berden, B., De Lange, B., (2003). Escape from the Tyranny of the Textbook: Adaptive Object Inclusion in AHA!. *Proceedings of the AACE ELearn 2003 Conference*, Phoenix, Arizona, November 2003, 65-71.

Freitas, V., Marcal, V. P., Gasparini, I., Amaral, M. A., Proenca M. L., Brunetto, M. A. C., et.al. (2002). AdaptWeb: An Adaptive Web-based Courseware, *Proceedings of the International Conference on Information and Communication Technologies in Education (ICTE)*, 20-23 Nov. 2002, Badajoz, Spain. (available on-line, accessed December 17, 2003 at http://www.inf.ufrgs.br/~palazzo/docs/Artigos/02 AdaptWeb ICTE (Badajoz).pdf.)

Kolb D. A. (1984). *Experiential learning experience as the source of learning and development*, New Jersey, Prentice Hall.

Mayer, R. E., Heiser, J., & Steve, L. (2001). Cognitive Constraints on Multimedia Learning: When Presenting More Material Results in Less Understanding. *Journal of Educational Psychology, 93* (1), 187-198.

Stash, N. & De Bra, P., (2003). Building Adaptive Presentations with AHA! 2.0. In the *Proceedings of PEG2003 conference*, Saint-Petersburg, Russia.

Wolf, C. (2002). iWeaver: towards an interactive web-based adaptive learning environment to address individual learning styles, online European Journal of Open and Distance Learning, (see www.eurodl.org). paper accessed, December 17, 2003 at http://www3.cti.ac.at/icl/archive/presentation/wolf.pdf.