

Adaptive features of a hypermedia system using bookmark per Web page

Hyosook Jung, Seongbin Park*

Department of Computer Science Education,
Korea University, Sungbook-Gu,
Anam-dong, Seoul, Korea
{est0718, psb}@comedu.korea.ac.kr

Abstract

In this paper, we present a technique called bookmark per Web page by which users can build a new link structure in a hypermedia. It allows users to create their own links for Web pages. Our aim is to provide an adaptive hypermedia with additional adaptive features by considering properties which the author of the hyperspace did not provide when the hyperspace was designed. In addition, a general hypermedia that does not have adaptive features can support personalized views of hyperdocuments using this technique. We expect that the bookmark per Web page can enhance information accessibility and reduce navigation disorientation and cognitive overload. We implemented our technique by modifying AHA! [De Bra *et al.*, 2003] which is an open source adaptive hypermedia system. We developed Java servlets and included them in the adaptation engine of AHA!. We added a link attribute for each concept at the user profile and assigned URL addresses of new links for its values.

1 Introduction

Adaptive hypermedia is a research area that deals with the problem of disorientation and cognitive overload possibly experienced by users of hypermedia systems by offering personalized contents and links based on a user model [De Bra *et al.* 1999]. While the navigation guidance is helpful [Brusilovsky, 1996], users may navigate with a rather passive manner because they read contents and follow links which the author of the hyperspace predicted and designed. In general, researchers on adaptive hypermedia concentrate on applications with fixed contents [Aroyo *et al.*, 2004] and it is assumed that the author knows the features of contents and the changes of user's properties well. However, it is difficult for an author to predict all changes of personal properties such as interests or aptitudes.

For example, some people may want to create direct links between local pages (i.e., *virtual*¹, internal links). Others may want to extend existing contents by adding links to sites which are found through information retrieval (i.e., *virtual*, external links). If users can create internal or external links according to their needs and preferences,

*To whom correspondence should be addressed.

¹We call it virtual because it does not exist in an adaptive hypermedia system, but a user creates it and adds it into the existing link structure.

they can obtain useful contents easily and quickly. Also, it is possible that people wish to create new navigational paths between local pages and to extend contents of other sites by using a search engine and this can be easily done with bookmark per Web page.

Figure 1 shows that user 1 and user 2 create internal and external virtual links individually during navigating information nodes which the author of a hyperspace designed. They can build personalized link structures according to their needs and preferences.

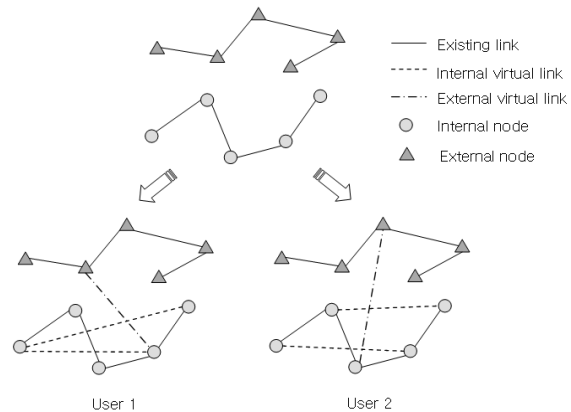


Figure 1: A link structure of a hypermedia and two different link structures created by different users

Bookmark per Web page allows users to create internal and external links within a current page according to their interests or needs. It presents individual links for each user on the page when they revisit the page. Therefore, users can jump between local pages or other sites. The link anchor and URL address are not written on the page. Information about links is extracted from the user profile the moment the requested page loads on the memory. We define a link which is created by the bookmark per Web page as a virtual link.

For the implementation of this technique, we modify an open source adaptive hypermedia system, AHA!. We include Java servlets in the adaptation engine of AHA! and the servlets perform creation and presentation of virtual links. AHA! has a number of special tags which can be used in a header or footer [De Bra *et al.*, 2003]. The tags are embedded with a `<object>` tag. They are defined in the standard "headerfooter.dtd". We insert a link anchor at the header. The anchor connects to a module that makes and modifies new virtual links. We present the virtual link

at the footer the moment users request a Web page.

This paper is organized as follows. Section 2 briefly describes related works. In section 3, we classify bookmarks into four types and compare them in various aspects. In section 4, advantages of using bookmarks per Web page in both an adaptive hypermedia system and a general hypermedia system are explained. In section 5, implementation of the technique in an adaptive hypermedia system as well as a scenario are explained. In section 6, implementation of the technique in a general hypermedia system as well as a scenario are described. Section 7 concludes the paper.

2 Related works

An adaptive hypermedia adapts the presentation of information and the overall link structure based on a user model [Brusilovsky, 1996]. An adaptive hypermedia uses adaptation techniques such as adaptive presentation and navigation. Authors decide on using different techniques to provide adaptation. An adaptive presentation technique includes conditional inclusion of fragments, stretchtext, etc. An adaptive navigation technique includes direct guidance, sorting of links, link annotation, link hiding, link disabling, link removal, Map adaptation, etc [De Bra *et al.*, 1999].

Several software platforms for adaptive hypermedia applications exist and AHA! is an open source general-purpose adaptive hypermedia system, through which very different adaptive applications can be created [De Bra *et al.*, 1999]. It consists of three models for adaptation. A domain model represents the contents of an application as concepts and concept relationships. A user model consists of attributes and their values to describe the user such as personal feature and knowledge about the domain. A user model is designed as an overlay of the existing generic domain model. An adaptation model is a set of adaptation rules that define how to update a user model and generate adaptive presentation and navigation. The adaptation rules are generically designed by the author [Wu *et al.*, 2001]. It is implemented in Java servlets which interact with three models. AHA! delivers the pages that correspond to links based on a user model when users request the pages by clicking the link anchors. It updates user's attribute values each time the user accesses a page.

3 Types of bookmarks

We describe four types of bookmarks based on four factors; where bookmarks are stored (storage), how bookmarks are maintained (management), which type of link is created (link type), whether relationships between created pages are clear (relationships). Table 1 shows the result of the comparison between types according to several factors.

Table 1: Comparison of three types of bookmarks : S(storage); M(management); L(link type); R(relationships) c(centralized); d(de-centralized)

type	[type1] bookmark in a browser	[type2] bookmark in a Web site	[type3] RDF Site Summary	[type4] bookmark per Web page
S	client	server	server	server
M	c	c	c	d
L	external	internal	both	both
R	indirect	indirect	indirect	direct

[type 1] is that a user creates bookmarks in a Web browser like Netscape or Internet Explorer. The bookmarks are stored at the client (i.e., user's computer). It is a centralized method that maintains a list of bookmarks in a folder or a directory. The bookmarks are URL addresses of Web sites.

[type 2] is that a user creates bookmarks when the user logs in a certain Web site such as an electronic newspaper or publishing sites. Users can select and see their interesting pages separately from other pages of the site. The bookmarks are stored at the server. It is maintained by a centralized method as a list which consists of all bookmarks in a management page. The bookmarks are URL addresses of internal links (i.e., local pages of the server).

[type 3] is a method that describes news or other Web contents which are available for distribution from an on-line publisher to Web users [Pilgrim, 2002]. They are stored at the server and maintained by a centralized method like [type 2]. It has the content on its site in the form of an RSS document. It can register contents of other Web sites as well as internal pages.

[type 4] is that a user creates bookmarks per certain Web page. The bookmarks are stored at the server. It is a decentralized method that maintains items related to each page. The bookmarks are URL addresses of external as well as internal links.

The next subsections compare the features of four types of bookmarks.

3.1 Storage

In [type 1], the workload of the server is small because the bookmark is stored at the client. However, if people do not use their own computers at home, they cannot use the bookmarks. Whenever their computers are changed, they have to import and export bookmarks. In [type 2,3,4], the workload of the server may be larger than [type 1] because the bookmarks of all users are stored at the server. But people can use the bookmarks everywhere at any time.

3.2 Management

[type 1,2,3] are maintained by a centralized method. That is a list of all bookmarks. As time goes on, the number of the bookmark items increases and some created bookmarks may not be used for months. Therefore, users must organize and maintain bookmarks such as naming folders and classify items [Abrams and Baecker, 1997]. In addition, they have to traverse many items of the bookmark list and recall the distinctive label of a bookmark whenever they want to revisit them. These extra-efforts may lead users to inconvenient situation. However, [type 4] is maintained by a decentralized method. (i.e., a small group of items per a Web page.) It makes links to relevant or interesting pages at current reading Web page. Without rummaging the items of the bookmark list, users can jump to a desired Web page or site at once and access information easily and fast.

3.3 Link type

The link type of [type 1] is an external link which is a Web site. The link type of [type 2] is an internal link which is a link between local pages of the server. However, users may want links to the contents of other sites when they search useful information. The link types of [type 3,4] are both internal and external links. It allows users to use relevant information at once.

3.4 Relationships

A Web page is represented as a concept for the contents of an application. [type 1, 2, 3] are a collection of individual concepts which do not have any relationships primarily. Users classify items with similar topic or manage a semantic hierarchy within folders or directories based on their aims or intentions. The concepts have indirect relationships, which means that pages or sites have similar topics. In [type 4], the concepts have direct relationships, which means that pages or sites are directly linked.

4 Using bookmark per Web page

Figure 2 shows the difference of navigation support and user's attitude between a general hypermedia and an adaptive hypermedia.

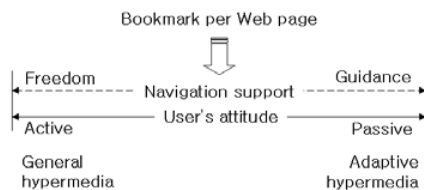


Figure 2: The difference of navigation support and user's attitude between general and adaptive hypermedia

A general hypermedia offers its users complete navigational freedom, but it can lead to comprehension and orientation problems. An adaptive hypermedia offers its users navigational guidance using adaptive links based on user model. An author of the application decides different guidance methods for adaptive links assuming that the author knows user's properties and contents well. An adaptive hypermedia tightly integrates freedom and guidance, but it is difficult for an author of a hyperspace to predict all possible properties of users in advance.

We use the bookmark per Web page that users create new links according to their interest, needs, goals, knowledge, etc. By using the technique, an adaptive hypermedia can perform adaptation, considering unpredicted changes of user's properties by giving user activeness. In addition, a general hypermedia can support personalized links based on user's interests and needs. In fact, bookmark per Web page can be considered as a simple open hypermedia functionality [Østerbye and Wiil, 1996].

4.1 Adaptive hypermedia

Adaptation techniques offer personalized contents and links that the author designed by predicting the changes of user's properties and actions. The properties or actions that the author did not predict can affect user's knowledge, interest, goal, etc. For example, if a user has low knowledge level on the adaptive hypermedia application, the user retrieves supportive information by a search engine when reading a Web page. It is helpful that there is a direct link to move to retrieved information at the current page when the user revisits the information. However, an adaptive hypermedia cannot recognize user's activities that occur at other applications. It cannot contain contents of other sites which are not under its control after finishing development.

To overcome these limitations, we present an adaptive hypermedia system to support bookmark per Web page which allows users to create new links at a Web page. If

each user can build individual link structures in a page according to their interests, needs and goals, they can access necessary information rapidly without passing through many nodes. Users can be provided abundant information conveniently if they can create internal and external links.

The technique can help to reduce disorientation because inefficient navigation can be avoided and jumping to the desired pages without wasting the capacity of working memory for recalling distinctive site labels becomes possible. It means that an adaptive hypermedia can support more elaborate adaptation than existing one by considering user's unpredicted properties. In addition, it means that people become both users and authors. In general, hypermedia authors just develop and modify contents and links while hypermedia users just read the contents and follow the links. By using bookmark per Web page, it can encourage users to understand hypermedia contents with an active and creative attitude.

4.2 General hypermedia

Generally, a bookmark is a saved link to a Web page that has been added to a list of saved links. When we are looking at a particular Web site or Web page and want to be able to quickly get back to it later, we can create a bookmark for it. In [Abrams *et al.*, 1998], they define bookmarks as personal Web information spaces to help people remember and retrieve interesting Web pages. The list that contains our bookmarks is the bookmark list. The bookmark in a Web browser or a Web site forces users to organize or maintain bookmarks for efficient retrieval but they tend to organize only when necessary [Abrams and Baecker, 1997]. It is because the items of the bookmark list are accumulated more and more as time goes on.

Also, users have to move to the bookmark list, remember distinctive labels, and look up a correct item. It is inconvenient for users although they manage the bookmark list well. It leads users to waste their working memory during reading. Instead of the centralized management method which gathers all bookmarks in one place, we apply decentralized management method which distributes bookmarks to each page. Although general hypermedia do not support adaptive techniques, bookmark per Web page allows users to have personalized information spaces.

5 Implementation in an adaptive hypermedia

We modify AHA! for applying the bookmark per Web page to adaptive hypermedia. Adaptation engine of AHA! consists of Java servlets that interact with a combined domain/adaptation model and with a user model [De Bra *et al.*, 2003]. It performs adaptation by executing the servlet which serves pages from the local pages or from pages of external site and updates the user model whenever each user visits a Web page [Wu *et al.*, 2001]. We include Java servlets that allow users to create internal and external virtual links and to present different virtual links of their own at the time they access a Web page.

Figure 3 shows the modified architecture of AHA! where new modules are added. We embed Java servlets in the adaptation engine of AHA!.

5.1 Creating virtual links

A link creator is to create and modify virtual links to AHA!. When a user wants to create virtual links at a Web page,

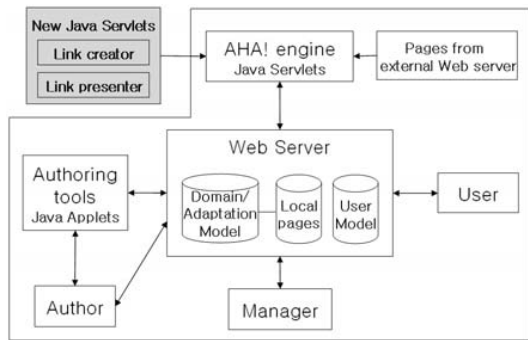


Figure 3: The modification of the architecture of AHA! by adding link creator and link presenter in the adaptation engine.

the user runs the link creator. Then, the user enters information about the current page and virtual link such as URL address, file names of local pages, or a concept of the current page. The link creator stores the information in his/her profile.

Table 2 is a part of a user profile related to a virtual link attribute. The user model is managed by XML files. The concept is “music.kandinsky” and the attribute value of its virtual link is “schonberg.xhtml”.

Table 2: virtual link attribute and its value in user profile

```
<record>
  <key>art.kandinsky.virtuallink</key>
  <type>string</type>
  <persistent>true</persistent>
  <value>schonberg.xhtml</value>
</record>
```

Note that contents of an AHA! application are documents written in XHTML. The header file is contained in XML files that are included in a page using the <object> tag. It must conform to the “headerfooter.dtd”. The DTD mentions the tags that are allowed in headers. It shows a personal attribute and its values to users such as login id, user name, course name, pages visited or non-visited, etc. Also it shows a group of tags called handlers. They create a link anchor through which users can access specially generated pages such as log out, knowledge setting, color setting, etc[De Bra *et al.*, 2003]. We include a link anchor which executes the link creator at a header of XHTML documents.

5.2 Presenting virtual links

We include a Java servlet on AHA!, which is called a link presenter. It presents individual virtual links for each user. When a user requests a certain Web page, the link presenter finds information of virtual links of the page from the profile of the user. Then it presents the virtual links at the footer of the page. Although users request a same page simultaneously, they can just see their own virtual links.

Like a header, a footer is an XML file that is included in an XHTML document using the <object> tag. The footer must conform to the “headerfooter.dtd” which mentions the tags[De Bra *et al.*, 2003]. Any HTML tags must be hidden inside <![CDATA[...]]> constructs because everything inside a CDATA sections is ignored by the XML parser. After

searching the information of a virtual link, the link presenter creates CDATA sections including the link anchor and URL address at the footer with HTML tags such as <A>. When AHA! parses an XHTML document, it presents the virtual link anchors with each URL address at the bottom of the document. That is, the URL address of the virtual link is assigned at ‘href’ attribute of <A> tag.

Table 3 shows that a virtual link to be inserted at the footer is included in a CDATA section. When the document is loaded, a link anchor to the document about Arnold Schonberg is created, “schonberg.xhtml”.

Table 3: A CDATA section including information of virtual link

```
<![CDATA [
  <A href="schonberg.xhtml">schonberg</A>
]]>
```

5.3 A scenario

We create an adaptive hypermedia application that offers learning contents about history of art and history of music. Even if each subject has special features according to the trend of the times, we can analogize relationships between two subjects because of human mental creature of the same times. The adaptive hypermedia application allows a learner to explore the close connections and affinities that exist between the two subjects. During learning the course, learners wish to link directly between local pages without passing through several nodes according to their needs or preferences.

For example, Kandinsky was inspired by works from Schonberg’s musical features, such as discarding chromaticism and abandoning tonal and harmonic conventions. Assume that a user is interested in Kandinsky’s complex relationship to Schonberg’s music. During browsing, the user finds separate documents about Kandinsky and Schonberg without a direct link. When reading a document about Kandinsky, the user creates an internal virtual link which jumps to a document about Schonberg. When revisiting the document about Kandinsky, the user can access interesting information easily by using the internal virtual link.

As another example, we assume that a user is interested in Renaissance Art. Because the period was characterized by a renewed interest in ancient Greek and Roman art and design, information about environment, science, and philosophy of the period are helpful for the user. The user retrieves relevant information by using a search engine and makes external virtual links to the local contents. The user can access abundant information rapidly by using the external virtual link.

Figure 4 is a screenshot that explains the life and works of Kandinsky. The user can use a link anchor, “Link Create” at the header. It executes a Java servlet that creates and modifies new virtual links.

Figure 5 shows a part of the document which is displayed when a user revisits it after creating a virtual link. The user can use a new link anchor with a button, “schonberg” at the footer.

Figure 6 depicts two different link structures that users may create in an adaptive hypermedia.

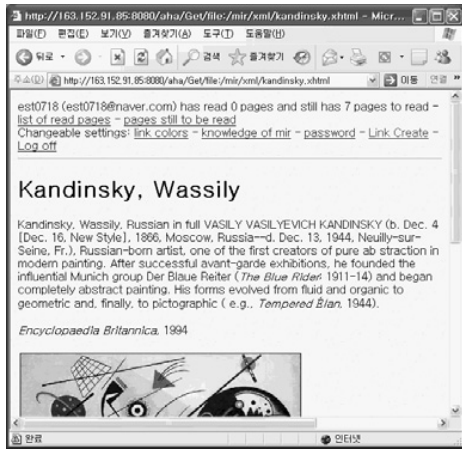


Figure 4: Creating virtual links using a Java servlet at the header



Figure 5: Presenting virtual links using Java servlet at the footer

6 Implementation in a general hypermedia

We apply the bookmark per Web page to a general hypermedia. Figure 7 shows a simple architecture for supporting the bookmark per Web page in general hypermedia which does not perform adaptation. A user logs in the Web server and requests a Web page. The request is sent to the virtual link extractor. The virtual link extractor searches virtual link information for the user at the profile repository and sends the information to the XHTML Handler. The XHTML Handler presents the user filtered page with virtual link anchor. When the user wants to create virtual links, the user runs the virtual link creator. The virtual link creator stores virtual link information at the profile repository.

6.1 Creating virtual links

When users want to create virtual links, they run virtual link creator and serve URL addresses of the current page and the virtual links to the virtual link creator. The virtual links can connect with internal and external pages. The virtual link creator stores the URL addresses at the profile repository which is a set of user profiles. In addition, users can modify and delete the added virtual links when necessary.

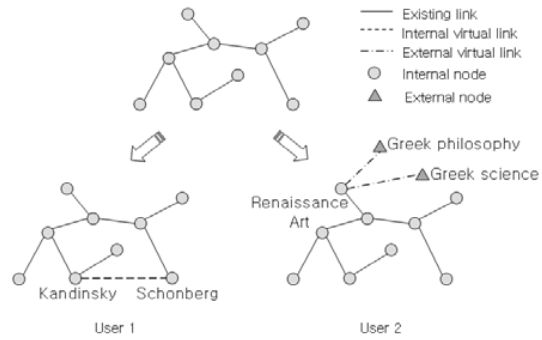


Figure 6: A scenario in adaptive hypermedia

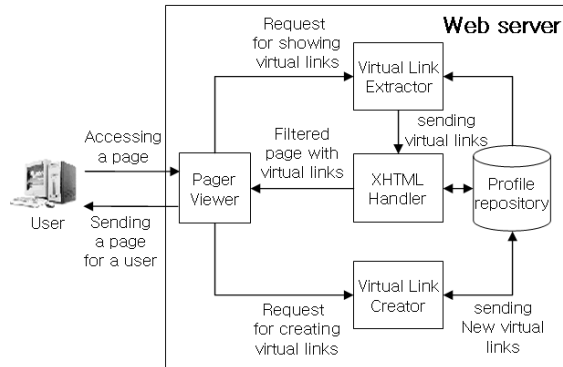


Figure 7: An architecture for the bookmark per Web page in general hypermedia

6.2 Presenting virtual links

When users access a page, the page viewer sends the virtual link extractor the request which is to show personal virtual links of the page. The virtual link extractor searches URL addresses of virtual links related to the requested page at the profile repository and sends the information to the XHTML handler which is a filtering module. When parsing the XHTML document, the XHTML handler includes extracted URL addresses on the memory which the document is being read. The page viewer presents virtual links in the lower end of the document.

6.3 A scenario

We create a general hypermedia application that offers learning contents about proteomics in Bioinformatics. Assume that a student reads a document about the processing of 3D structure data of large molecules of proteins. Assume also that the student has low knowledge level about biology and the student retrieves basic information about protein structure and function by search engines. The student creates external virtual links to the retrieved pages which contain relevant information. When revisiting the document about processing of 3D structure data, the student can access to supportive retrieval information rapidly.

We assume that another student studies contents about proteomics and reads a document about algorithms to protein structure comparison. Because the student needs to frequently refer to the document about the processing of protein structure data to understand the algorithm, the student creates an internal virtual link between two documents.

Figure 8 depicts two different link structures that users can create in a general hypermedia.

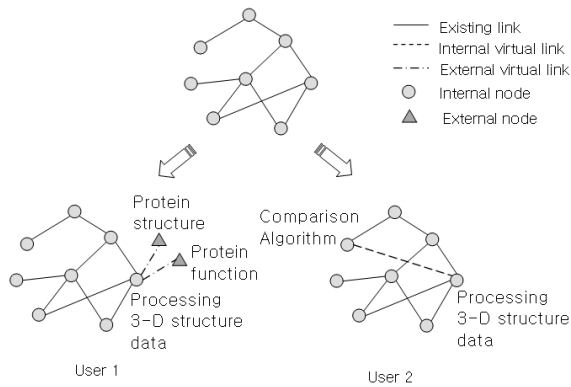


Figure 8: A scenario in general hypermedia

7 Conclusions

This paper presents a technique which creates bookmarks per Web page. It provides a hypermedia with additional adaptive features by considering user's properties that authors do not predict because users can create new links according to their needs or preferences. It can enhance information accessibility because it helps to reduce inefficient browsing activities through personalized link structures.

The bookmark per Web page has several advantages. First, users can use the bookmark everywhere at any time because it is stored at the server. Second, users can jump to a desired page or site at once without searching the list of bookmarks because it is maintained by a decentralized method. Third, it can help the extension of existing information because it supports both internal and external links. Forth, it can change learner's attitude from passive to active state since users can create links.

We note that because users are responsible for link creation, the usefulness of the bookmarks per Web page can increase when users have basic knowledge levels on the subject so that they know which Web pages that the author of a hyperspace do not connect may look helpful once connected.

An added Web page means a new concept for the contents. If a user adds a new bookmark at a certain Web page, we can infer that the user is interested in or has knowledge about the concept which is related to the Web page. We are currently investigating ways by which adaptation functionality of adaptive hypermedia can be enhanced by updating interest or knowledge values of the user model when users add a bookmark per Web page.

8 Acknowledgement

We wish to thank the anonymous reviewers for their helpful comments and suggestions.

References

- [Abrams and Baecker, 1997] David Abrams and Ron Baecker. How People Use WWW Bookmarks. *CHI 97 Electronic Publications: Late-Breaking/Short Talks*, March, 1997.
- [Abrams et al., 1998] David Abrams, Ron Baecker, and Mark Chignell. Information Archiving with Bookmarks: Personal Web Space Construction and Organization. *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 41-48, 1998.

- [Aroyo et al., 2004] Lora Aroyo, Paul De Bra, Geert-Jan Houben, and Richard Vdovjak. *Embedding information retrieval in adaptive hypermedia: IR meets AHA!*. *Hypermedia*, 10(1):53-76, 2004.
- [Brusilovsky, 1996] Peter Brusilovsky. *Adaptive Hypermedia: An Attempt to Analyze and Generalize*. Lecture Notes in Computer Science, 1077:288-304, 1996.
- [Conklin, 1987] Jeff Conklin. *A Survey of Hypertext*. *IEEE Computer*, 20(9):17-41, Sept, 1987.
- [De Bra et al., 1999] Paul De Bra, Peter Brusilovsky, and Geert-Jan Houben. *Adaptive hypermedia: from systems to framework*. *ACM Computer Survey*, 31(4es), 1999.
- [De Bra et al., 1999] Paul De Bra, Geert-Jan Houben, and Hongjing Wu. AHAM : A Dexter-based Reference Model for Adaptive Hypermedia. *HYPERTEXT '99: Proceedings of the tenth ACM Conference on Hypertext and Hypermedia : returning to our diverse roots*, pages 147-156, September, 1999.
- [De Bra et al., 2003] Paul De Bra and Natalia Stash and Barend de Lange. AHA! Adding Adaptive Behavior to Websites. *Proceedings of the NLUUG Conference*, May, 2003.
- [Østerbye and Wiil, 1996] Kasper Østerbye and Uffe Kock Wiil. The Flag Taxonomy of Open Hypermedia Systems. *HYPERTEXT '96: Proceedings of the seventh ACM Conference on Hypertext and Hypermedia*, pages 129-139, 1996.
- [Pilgrim, 2002] Mark Pilgrim. *What is RSS?*. <http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html>, December 18, 2002.
- [Wu et al., 2001] Hongjing Wu, Erik de Kort, and Paul De Bra. Design issues for general-purpose adaptive hypermedia systems. *HYPERTEXT '01: Proceedings of the twelfth ACM conference on Hypertext and Hypermedia*, pages 141-150, 2001.