

# User-Controlled Link Adaptation

Theophanis Tsandilas  
Dept. of Computer Science  
University of Toronto  
Toronto, Canada  
fanis@cs.toronto.edu

m.c. schraefel  
Dept. of Electronics and Computer Science  
University of Southampton  
Southampton, UK  
mc@ecs.soton.ac.uk

## ABSTRACT

This paper introduces an adaptable hypermedia approach applied to adaptive link annotation techniques. This approach suggests that the combination of direct manipulation with automated link annotation affords greater user control over page adaptation. In turn, this direct control better supports user focus in information discovery tasks. Unlike adaptive-only systems, our approach lets users both define multiple topics of interest and then manipulate how these topics' associated links are presented in a page. We discuss how the approach can be applied both to pages viewed as well as to the user's history list, thereby relieving users from the task of either adding to or organizing bookmarks. We describe the prototype developed to support these manipulations, as well as the adaptive architecture developed to support these controls.

## Categories and Subject Descriptors

H.5.4 [Information Interfaces and Presentation]: Hypertext/Hypermedia – *Navigation*. H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – *Information Filtering*.

## General Terms

Design, Human Factors

## Keywords

Adaptable hypertext, direct manipulation, navigation assistance, hyperlink annotation, history visualization

## 1. INTRODUCTION

Current browsers provide little support for helping users to locate useful information within web pages and decide which links to follow. The discovery of interesting information requires the reader to scan pages thoroughly and possibly follow several links before identifying material relevant to his or her interests. This process may cause frustration and even disorientation to the user. Conklin [8] identifies *disorientation* and *cognitive overhead* as the main problems that users experience when trying to navigate within hypertext systems. Nielsen [20] observed that even in small hypertext systems, users can lose their orientation if no orientation clues are provided. Cognitive overhead is defined by Conklin as the “additional effort and concentration necessary to maintain several tasks or trails at one time” [8] and has been associated with the limited capacity of the human's short-term memory [12].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HT'03, August 26-30, 2003, Nottingham, United Kingdom.

Copyright 2003 ACM 1-58113-704-4/03/0008...\$5.00.

It can be caused when the user is given a large number of choices and needs to make decisions about which links to follow and which to abandon. Humans do not have the cognitive capability to evaluate and process all the available Web information. Empirical results [32] associate disorientation and/or cognitive overhead with the number of links that appear in hypermedia systems, showing a reduction of reading and learning performance as the number of links increases. These results suggest that link-filtering tools can be useful.

Although designing pages that structure information and provide semantic relationships between topics can assist navigation [21], user interests do not always match the information structure that the author of a page imposes on its viewers. At the same time, a large part of the Web does not obey any design guidelines or structure. Furthermore, a single browsing session may involve sites and pages that employ different conceptual models to organize information disturbing the *coherence* of the user's mental model [27].

There has been some prominent effort towards developing utilities that assist navigation in the Web. One such utility is bookmarks, which allow for the creation of personalized web spaces by organizing pages according to user interests. However, bookmarks engage several problems: (1) the user has to locate and assess an item before adding it to the bookmarks list; (2) their maintenance requires effort from the user; and (3) items are categorized under fixed categories, so they cannot be associated with multiple and evolving user interests. Another utility is search engines, which allow users to specify their interests and receive automatically generated lists of related documents. The collective documents from search results can be considered as user-determined views of the Web. Unfortunately, these documents are isolated from their hypertext structure and the surrounding context.

Adaptive hypermedia systems [5] have tried to reduce the gap between user interests and information in hypertext by providing navigational support according to a user model, which captures information about a user's interests and goals. This approach suggests that disorientation and cognitive overhead can be reduced by annotating, hiding, organizing or recommending links with respect to the user's information needs. The main criticism against adaptive systems is that they do not allow direct user control over the adaptation process [25]. User models are persistent or change slowly, and their construction is based on assumptions that do not always hold. An example assumption is that user interests can be inferred from interests that were expressed in the past by the same or similar users. Further frustration is induced when the system makes incorrect guesses about what the user wants.

In this paper, we present a prototype that incorporates direct manipulation and adaptive feedback into Web browsing. Users can explicitly specify the degree to which different topics interest

them via interface manipulators implemented as sliders. Links in the browsed pages are automatically adapted based on their relevance towards the weighted topics specified by the sliders. Changes in topic interests are instantly reflected on the pages that the user views. We propose adaptation techniques based on link annotation that enable smooth and continuous adaptation effects, which help the user comprehend the adaptation mechanism and use the sliders effectively to identify interesting links. In addition to controlling the presentation of links on the pages, the sliders allow the user to discover and reassess pages and links that appeared in the past connecting them to user interests by means of an interactive visualization tool that displays overviews of the browsing history. Since analysis and archiving of visited pages are performed automatically, the user is relieved from the tasks of adding and organizing bookmarks.

Altogether, our prototype can be considered as a hybrid between an adaptive and an adaptable hypertext system. It is adaptive, since it automatically adapts the presentation of linking information based on the specified user interests. On the other hand, it is adaptable, since interests on which adaptation is based are directly controlled by the user.

The rest of the paper is structured as follows. Section 2 discusses related work and stresses the contribution of our approach. Section 3 presents an overview of our prototype and describes its main components. Issues concerning the representation of user interests and pages as well as the measures used to calculate relevance between interests and pages are examined in Section 4. Section 5 focuses on techniques exploited by our prototype to adapt links within Web pages. Related techniques used by other systems are also discussed. Section 6 presents our approach on visualizing and accessing the browsing history. Finally, Section 8 summarizes the paper and identifies future work.

## 2. RELATED WORK

Providing navigation support with respect to user interests has been the goal of several adaptive systems. WebWatcher [15] highlights a small number of hyperlinks on each page based on a set of keywords specified by the user at the beginning of the browsing session. Although users can explicitly specify their exploration interests, multiple topics of interest cannot coexist. Another weakness of this system is that switching between two different topics requires starting a new session and providing a different set of keywords, a procedure which requires a lot of user effort. Other systems learn user interests from explicit user ratings [22], by analyzing the content of previously visited pages [19] or by making various observations about the user's behaviour [18]. The main drawback of the above systems is that they maintain persistent and slowly-changing user models overlooking the fact that different browsing sessions of the same user or even a single session may engage different user interests and goals. Techniques that extract the context [13; 17; 3; 9] of a browsing process can accommodate multiple user interests. However, switching between interests is out of the user's control. The user may not even be aware of the adaptive behaviour of the system or its belief about his or her browsing context.

Other approaches provide interaction techniques to aid the browsing task of a user and reduce the cognitive overhead associated with the task of deciding whether a link is related or not. Fluid links [31] annotate links with additional information about the destination page so that the user can decide on how worthwhile a hyperlink is before viewing the destination page.

Annotations are inserted below a link anchor after the user positions the mouse over it, while animation is used to provide natural transitions in the user's view. In the same context, Link Lens [26] exploits the metaphor of magic lenses to help the user uncover information about a link, while HyperScout [29] uses *pop-ups* to display additional information. The main drawback of these techniques is that users are still required to hover over link anchors and read the attached annotation before deciding on the relevance of the destination pages. In other words, no automated guidance is provided that would help the user to locate interesting links without the need of reading and evaluating additional information.

Display of additional information initiated by *hovering gestures* is also supported by ConTexts [24], a model for designing adaptable hypermedia systems. In addition, the ConTexts approach allows the user to manipulate the displayed content on a page and change the reading focus by expanding and collapsing text segments. Again, the model does not support any mechanism that could help the user to directly locate interesting information, for example, by automatically expanding relevant text segments and collapsing irrelevant ones.

Direct manipulation, on the other hand, has only been applied in specific application domains, where information is structured and manipulators correspond directly to data descriptors. A representative example of direct manipulation is FilmFinder [2], which allows users to query and filter visualized information about movies by means of simple interface manipulators, such as sliders. Since the Web involves unstructured information and a large number of different domains, our approach considers manipulators as controllers of user interests rather than controllers of specific data descriptors.

On the other hand, several approaches have proposed structuring and visualization techniques to assist users in returning to previously visited pages. Cockburn and Greenberg [6] identify four categories of techniques that support the user's navigational task by organizing the display of pages: (1) hub-and-spoke dynamic trees that depict the user's navigational branching actions; (2) spatial layouts that exploit the user's memory for the spatial location of objects; (3) site-maps that exhibit the physical storage location of pages; and (4) temporal organizations that exploit the user's memory for the timing of their actions. Only the second category of techniques allows for structuring previously visited pages according to user interests. However, these techniques require the user to explicitly select and organize the pages. Since the structures are static, the pages cannot be associated with dynamically changing user interests. In a study by Abrams et al. [1], 37% of the surveyed users declared that they did not organize bookmarks, as organizing bookmarks is a laborious and time-consuming task. Our approach overcomes this problem by automatic categorization of pages based on the specified topics of user interests. Pages can be associated with multiple topics and their retrieval is facilitated by interface manipulators, which act as filtering tools.

## 3. SYSTEM OVERVIEW

This section presents an overview of the prototype and describes its main components. Combining page adaptation with direct manipulation is achieved by the collaboration of two main components. The first is the *adaptation controller*, which resides at the client side, and allows the user to specify topics of interests

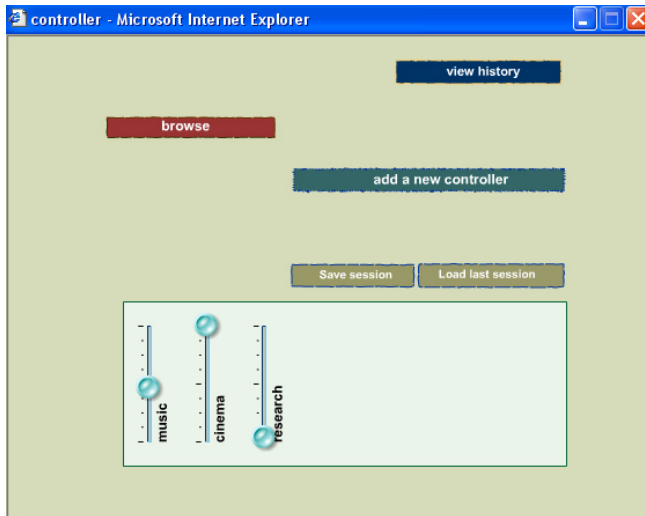


Figure 1. The adaptation controller

and control the adaptation of the pages by means of interface manipulators. The adaptation controller also provides tools for viewing and controlling the browsing history. The second component is the *page modifier*, which resides at the server side and is responsible for handling Web page requests. The page modifier modifies pages requested by the user into pages that can be dynamically adapted by the adaptation controller. Browsing of pages is performed by using a common Web browser. We continue with a more detailed description of the two components.

### 3.1 Adaptation Controller

The adaptation controller is presented in Figure 1. It contains a panel of sliders, which correspond to different topics of user interests. By moving a slider up and down, the user can specify whether and to which degree the corresponding topic describes the current browsing process. Five buttons provide additional functionality. The first button (“view history”) enables the display of the visualization component allowing the user to overview pages that have appeared in the past. This component is discussed in more detail in Section 6. The second button (“browse”) activates a new panel, which lets the user request the display of new pages. The third button (“add a new controller”) displays a panel for adding new topics of interest that result in new sliders. An instance of this panel is presented in Figure 2. As shown in the figure, the user is asked to enter a name for the slider and a set of keywords that define the corresponding topic. New sliders can be added dynamically during a browsing session. Finally, the last two buttons (“Save session” and “Load last session”) can be used to save the current session and load the last saved session, respectively. Saving concerns both the browsing history and the definitions of topics of interest.

The adaptation controller is implemented in Flash MX. Communication with other windows of the browser is handled by JavaScript code.



Figure 2. Adding a new slider

### 3.2 Page Modifier

The page modifier is responsible for handling requests for new pages. The whole component is implemented in Java using Servlet technology. Page requests are activated either by following links displayed in the browser or by using the browsing and history panels in the adaptation controller. Additional requests are activated from the adaptation controller in order to exchange information with the page modifier. More precisely, whenever a new slider is added, the associated information, i.e., the keywords describing it, is sent to the page modifier by means of a POST HTTP request. In a similar way, the adaptation controller receives information about the history of a browsing session.

Requests for Web pages are sent to the page modifier by incorporating the URL of the requested page into the URL space of the servlet. The page modifier fetches the requested page and extracts all the source (SRC) and link (HREF) anchors in the page. Relative URLs in extracted anchors pointing to non-textual documents, e.g., images, are transformed to absolute URLs. Anchors pointing to HTML and text pages are handled differently. First, the content of the destination page is analyzed and a vector representation of the containing text is created. The vector representation of the page is compared to the vector representations of the user interests represented by the sliders. The result is a list of values denoting relevance between the content of the page and the corresponding user interests.

The final adaptable page is constructed by modifying the HTML code of the original page as follows. Each URL pointing to non-textual documents is replaced by the corresponding absolute URL. URLs pointing to HTML and text pages are replaced by the URL of the page modifier’s servlet accompanied by the absolute URL of the original page. For instance, assuming that the URL of the original page is <http://www.utoronto.ca>, the URL that is finally constructed is:

<http://localhost:8080/Modifier?URL=http://www.utoronto.ca>

In this way, following hyperlinks on the browser results in page requests that are served by the page modifier. In addition to translating the containing URLs, link anchors referring to HTML or text pages are assigned identifiers allowing JavaScript functions to apply adaptive annotation to the anchored text when displayed on the browser. Depending on the adaptation technique that is applied, parameters in the anchor’s tag that explicitly define the presentation style are removed. Adaptation techniques are discussed in Section 5. As a final step, the response is enhanced with JavaScript code providing the values of relevance between the linked pages and the topics that the user has defined in the adaptation controller. This JavaScript code allows the sliders on the adaptation controller to modify the style of the linked text on the rendered page the with respect to the calculated relevance values.

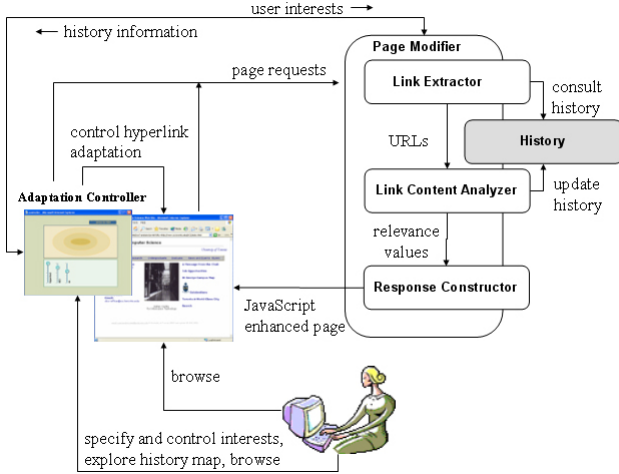


Figure 3. System overview

An overview of the system is presented in Figure 3. As shown in the figure, the page modifier contains a history of previously analyzed pages in the form of vector representations. Before accessing a page addressed by a given URL, the page modifier checks whether this page has been accessed and analyzed in the past. If this is the case, the page is not accessed again. In other words, the history acts as a cache memory, which speeds the process of building the response. The history is also the source of information for the visualization component that resides in the adaptation controller. As a browsing session may involve a large number of pages, the history is pruned after history information has been transferred to the adaptation controller.

## 4. USER INTERESTS AND DOCUMENTS

### 4.1 Representation and Relevance Metrics

We represent both user interests and documents, i.e., the content of Web pages, as vectors of terms, which are usually referred as *features* in Information Retrieval. Each feature in a vector is a stemmed version of a single word and is assigned a real value. This value is calculated by means of the widely used TFIDF heuristic [23]:

$$TFIDF(w, d) = TF(w, d) \cdot \log\left(\frac{n}{DF(w)}\right) \quad (4.1)$$

where  $TF(w, d)$  is the frequency of the feature  $w$  in document  $d$ , and  $DF(w)$  is the number of documents that contain  $w$  in a total of  $n$  documents.

The generation of large feature vectors is achieved by eliminating stop-words and omitting features with very small TFIDF values. In the current prototype, topics of interest are defined as sets of keywords, manually entered by the user. However, the above representation allows for automatic extraction of user interests, e.g., by clustering the vector representations of pages that the user has visited or bookmarked in the past [9]. In this case, different topics of user interests could be represented by the centroids of the resulting clusters.

We use the cosine similarity metric [23] to measure relevance between topics of user interests and documents. More precisely, the relevance between the  $i^{th}$  interest topic and a document  $d$  is expressed as the cosine of the angle  $\omega_i$  between their vectors:

$$\cos(\omega_i) = \frac{\vec{v}_i \cdot \vec{d}}{\|\vec{v}_i\| \cdot \|\vec{d}\|} \quad (4.2)$$

where  $\vec{v}_i$  is the vector representation of the interest topic,  $\vec{d}$  is the vector representation of the document, and  $\|\cdot\|$  stands for the Euclidean norm.

### 4.2 Combination of Multiple Interests

As discussed in Section 3, each interest topic is controlled by an individual slider. A slider defines the degree to which the corresponding topic expresses the interests of the user. User interests can involve more than one topic. The significance of each topic in the user interests can vary as each slider defines a different weight for the topic that it represents. However, it is not evident how the weights on the sliders should be interpreted by the user and accordingly by the system. We distinguish between two useful interpretations of a particular instance of slider weights.

#### 4.2.1 OR Interpretation of Slider Weights

The first interpretation, which is the one that our current prototype adopts, assumes that the user is looking for documents that relate to any of the highly weighted topics of interest, independently to each other. Documents that relate to many highly weighted topics are assumed to be more relevant than the ones that relate to few topics. For instance, an assignment of values 1, and 0.5 for the topics *music* and *cinema*, respectively, would result in high relevance scores for pages with many occurrences of terms that relate to music or cinema. As music has a higher weight, a page with 20 occurrences of terms relating to music and 10 occurrences of terms relating to cinema would get a lower relevance score than a page with 30 occurrences of terms relating to music and without any occurrence of words that relate to cinema. In this case, the relevance between a document  $\vec{d}$  and a set of weighted topics  $I$  with size  $|I|$  can be expressed as the average of the weighted values of relevance between the document and the individual topics:

$$relevance(I, \vec{d}) = \frac{1}{|I|} \sum_{i=1}^{|I|} w_i \cos(\omega_i) \quad (4.3)$$

We denote by  $w_i$  the weight of the  $i^{th}$  topic of user interests, which derives by normalizing the weight of the corresponding slider, so that  $0 \leq w_i \leq 1$ .

The calculation of the above expression is performed by the system as follows. First, the page modifier calculates all the individual terms  $\cos(\omega_i)$ , according to Equation (4.2). The resulting values are integrated into JavaScript statements within the modified page that the page modifier sends to the browser. The final calculation of Equation (4.3) is performed by a JavaScript function based on the weights on the sliders. The calculation is repeated whenever a weight changes.

#### 4.2.2 AND Interpretation of Slider Weights

The second interpretation assumes that the user is looking for documents that relate to all the highly weighted topics of interest. In this case, an assignment of values 1, and 0.5 for the topics *music* and *cinema*, respectively, would result in high relevance scores for pages whose main topic is music, while their secondary topic is cinema. According to this interpretation, a page with 20 occurrences of terms relating to music and 10 occurrences terms relating to cinema could get a higher relevance score than a page with 30 occurrences of terms relating to music but without terms

that relate to cinema. This suggests that relevance should be computed as follows:

$$\text{relevance}(I, \vec{d}) = \cos(\omega_I) = \frac{\vec{v}_I \cdot \vec{d}}{\|\vec{v}_I\| \cdot \|\vec{d}\|} \quad (4.4)$$

where  $\vec{v}_I$  is the vector that results when summing the weighted vectors of the individual topics:

$$\vec{v}_I = \sum_{i=1}^{|I|} w_i \vec{v}_i \quad (4.5)$$

and  $\omega_I$  is the angle between the above vector and the document vector. By combining the two previous equations with Equation (4.2), we get the following expression:

$$\text{relevance}(I, \vec{d}) = \frac{1}{\|\vec{v}_I\|} \sum_{i=1}^{|I|} w_i \|\vec{v}_i\| \cos(\omega_i) \quad (4.6)$$

As the weights  $w_i$  cannot be pushed out of the Euclidean norm of  $\vec{v}_I$ , vectors representing topics of interest cannot be isolated from the final calculation of the relevance value. Additional parameters, also included as JavaScript statements within the page modifier's response, are required to describe this norm. In the case of small vectors, i.e., when interests are defined by small numbers of keywords, only a small number of additional parameters are needed.

## 5. DIRECT LINK ADAPTATION

### 5.1 Link Adaptation: Historical Context

Brusilovsky [5] identifies five main types of adaptive navigation support: direct guidance, adaptive link sorting, adaptive link hiding, adaptive link annotation, and map adaptation. With the exception of map adaptation, different combinations of the above techniques have been employed by adaptive systems to direct the navigational tasks of users or assist them in discovering interesting information. WebWatcher [15] and Personal WebWatcher [19] apply direct guidance by highlighting a small number of links as the most relevant to the user interests. Syskill & Webert [22] visualizes multiple degrees of suggestions by annotating links with an icon indicating likelihood of user preference together with the estimated probability that the user would like the target page. Other systems [16] apply link sorting to suggest lists of pages in order of relevance to the current user interests.

All the above adaptation techniques were selected under the perspective that the user could not directly control the adaptation process. Thus, they cannot be used to reflect continuous changes in the user's interests. Continuous changes in the user's interests should be accompanied by appropriate visual feedback, so that the user can predict the adaptation mechanism. Direct guidance and link hiding do not respect this requirement since they use only two different states to represent and display links. Adaptive link sorting allows multiple, although discrete, number of states, but its application is limited in reordering lists of links. Furthermore, multiple changes in the ordering of links may result in user confusion, especially when the original sorting reflects a particular conceptual order, for example, when links are alphabetically sorted. For these reasons, we decided to adopt link

annotation techniques to achieve continuous and non-intrusive adaptation. Using the visual characteristics of the anchored text, i.e., the colour and the size of fonts, is a common technique for annotating links. For instance, colour is employed by common browsers to distinguish between regular links and links that have been previously visited. Hypadapter [14] applies link annotation by using multiple levels of font sizes.

### 5.2 Direct Manipulation and Link Adaptation

We tested several combinations of using colour and font sizes to represent different degrees of relevance between links and user interests. Figure 4 presents an example of link adaptation by adjusting the font sizes of the links. More relevant links are presented with large fonts while irrelevant links are presented with small fonts. In the figure's example, a university page has been visited with many links pointing to information about students, courses, housing, etc. Three sliders have been created that correspond to three different topics: courses, housing, and faculty, defined by a small number of keywords. As shown in the figure, the user has specified interest in information about housing. As a result of this, links pointing to pages about housing and residences are highlighted by larger fonts. Font sizes change smoothly as the user manipulates the sliders. This allows smooth transitions between interests without disturbing the view of the user. By moving the sliders and testing different combinations of values, the user can identify the mapping between interests and links, and comprehend the underlying model that the system employs to suggest links.

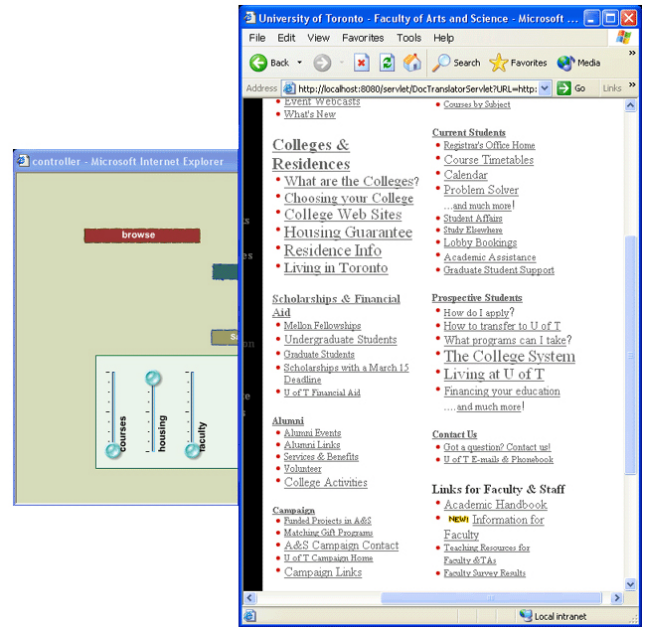
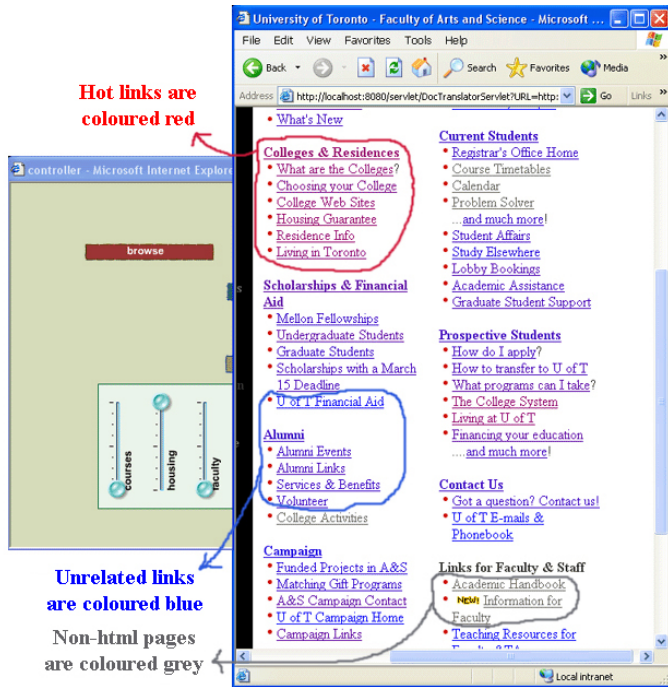


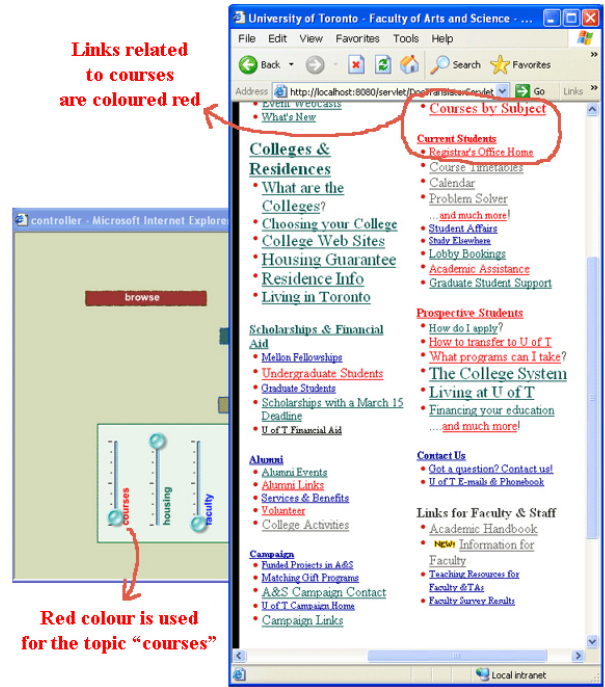
Figure 4. Adapting the font size of the links

Figure 5a presents the same example, but colour instead of font sizes is used to annotate links. Relevant, "hot" links are coloured red, while irrelevant links are coloured blue. Links that point to documents such as PDF files, the content of which is not recognized by the system are annotated by using a grey colour. By changing the balance between red and blue, different degrees of





(a) Adapting the hue of the colour between red and blue



(b) Applying different colours to different topics

Figure 5. Adapting the colour of the links

relevance can be represented. Again, the transition between interests is continuous and smooth. Figure 5b shows how adaptation of font sizes and colours can be combined to provide additional information to the user. Colours, now, represent different topics of user interests. The colour of a link derives from the topic that is closer to the content of the destination page and is independent of the slider values, which only control the font sizes. The mapping between colours and interests is manifested by the colour of the labels on the sliders. This technique helps the user identify different topics on the page before even starting manipulating the sliders. Its main drawback is that it can only apply when a small number of topics exist, as the use of more than 5-6 colours [30] can confuse the user and decrease his or her performance. However, the problem relaxes if only a small number of topics are allowed to be shown on a single page, e.g., the most dominant topics or the topics that correspond to the current user interests.

Adaptive link annotation by adjusting the font sizes can be viewed as a fisheye-view technique. Fisheye views [10] support the presence of multiple focal points and different levels of zooming at the same time. Bederson [4] applied fisheye zooming to pull-down menus with the goal to reduce the cognitive load caused by long lists of choices. His pilot study showed that some users found the technique helpful. Although our goal is also to reduce the cognitive load caused by long lists of choices, we apply fisheye zooming differently. In our case, the point of focus is not determined by the position of the mouse's cursor but rather by the user's interests as defined by the sliders on the adaptation controller.

Fisheye-view techniques define a *Degree of Interest (DOI)* function which specifies how the elements of the visualization are presented, e.g., their size, based on the current point of focus. In

our case, the DOI function is defined by the relevance function as presented in Equations (4.3) and (4.6), and determines either the font size or the colour of the links on a page. In order to produce uniform distributions of values of font sizes and colour ranges, the actual DOI function is expressed as a logarithmic function of the relevance function:

$$DOI = \max + a * \log(\text{relevance}) \quad (5.1)$$

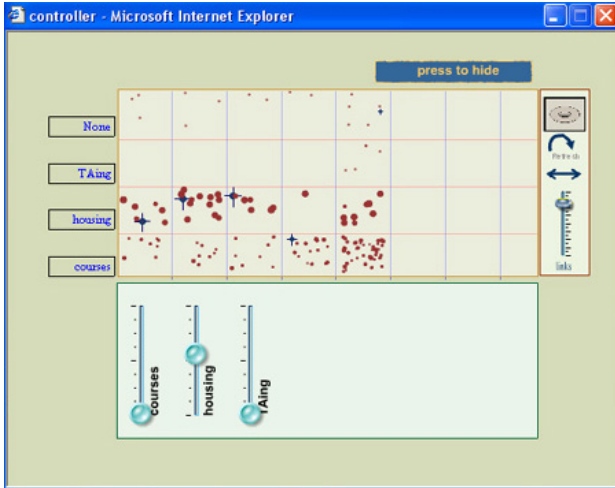
where  $\max$  is the maximum value of the allowed font sizes or colour ranges, and  $a$  is a constant, which determines the sensitivity of the visualization to relevance variations.

## 6. BROWSING HISTORY

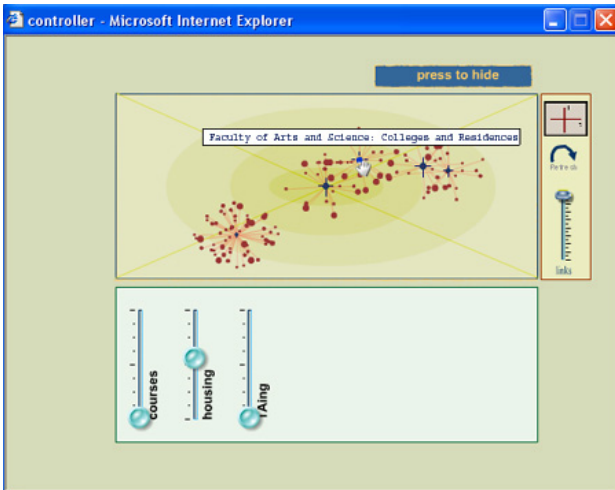
### 6.1 Overview

In addition to links on pages shown on the browser, our approach applies user-controlled adaptation to history lists. History lists and bookmarks supported by existing browsers facilitate the access of previously visited pages. However, these utilities are largely unused due to the effort that their use requires. Previous research [7] has stressed the need for techniques that automatically capture information about the pages that the user visits and relieve the user from the process of spending time maintaining bookmarks. The title and the URL of the page, its access time and sometimes a thumbnail [7] form page descriptions that existing history tools capture and use to organize browsing histories. This approach, however, offers limited searching capabilities and assumes that the user remembers the above-mentioned page descriptors in order to be able to access a page.

Our system captures rich page descriptions in the form of feature vectors, which enable the use of more sophisticated history organizations and searching mechanisms. More specifically, according to our approach, pages are automatically associated



(a) X-Y visualization organizing nodes by time and topic



(b) Circular visualization organizing nodes by time

**Figure 6. History visualizations**

with multiple topics based on their actual content. The organization of the pages can change dynamically based on the topics that the user introduces and manipulates by means of the sliders in the adaptation controller. Not only information about previously visited pages but also information about pages linked

by them is captured by the system. The rationale of this approach is that interesting pages are usually surrounded by other interesting pages, and following a link from a certain page is often succeeded by returning to the same page and following a different neighbouring link. Also, the significance of a page highly depends on the significance of its links. In other words, a page which is not very relevant by itself but contains many relevant links may be more useful than a relevant page with no relevant links. Therefore, providing information about the links that reside in a page can help the user decide about its importance.

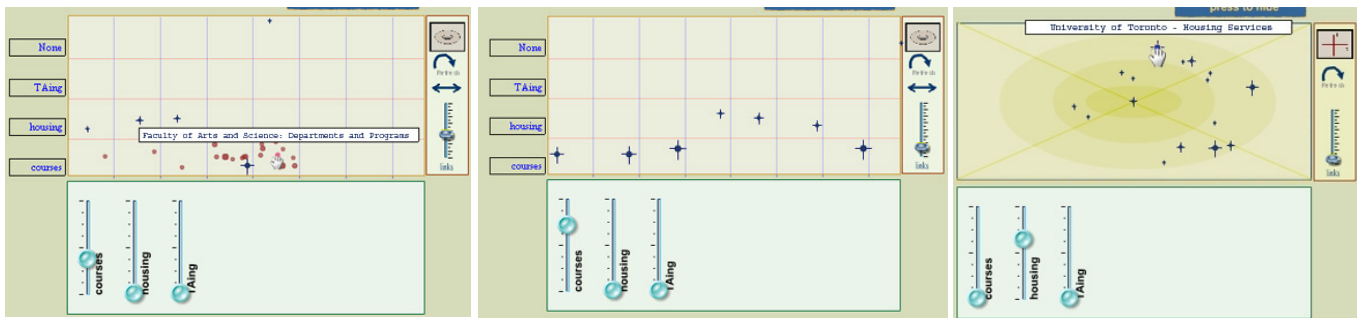
Our approach can be viewed as a map adaptation technique. Map adaptation, which refers to techniques that adapt the form of global or local hypermedia maps presented to the user [5], has not been adequately supported by existing adaptive hypermedia systems. Our approach provides global maps of the browsing history, where adaptation is performed by highlighting or filtering displayed items with respect to the user's interests.

## 6.2 History Visualizations

We describe two visualizations afforded by our prototype. Figure 6a presents the default visualization which organizes visited pages and their links by two dimensions: the visiting time and the topic that is most relevant to the page or the link. Each column in the visualization area corresponds to a single page visit. Columns at the left part of the visualization correspond to the most recent visits. Each row, on the other hand, corresponds to a different topic of user interests. There is an additional row for pages and links that are irrelevant to any of the available topics. Visited pages are denoted by crosses, while links are denoted by bullets.

The distribution of the visualized nodes over different topics allows the user to draw conclusions about the content of previously visited pages and their relevance to different topics. Further observations can be made by manipulating the sliders at the bottom of the adaptation controller. The size of the nodes in the visualization is adjusted by the values of the sliders allowing users to identify which nodes relate to their current interests. Both nodes representing previously visited pages and nodes representing links are interactive. By hovering over a node, information about the corresponding page, e.g., the title of the page is displayed. Clicking on a node sends an HTTP request to the page modifier to display the corresponding page on the browser.

The main drawback of the first visualization technique is that only a small part of the history can be displayed at a certain moment. Moving back or forward in the history is performed by clicking



(a) Only links relevant to courses are displayed

(b) Links are hidden - The change of topics over time is clearly shown

(c) Links are hidden - The whole space of visited nodes is shown

**Figure 7. Filtering and hiding the links in the visualizations**

the arrows in the *control panel* that appears at the right of the visualization area. In their study on bookmarks, Abrams et al. [1] suggested that visualizing a large number of items at the same time instead of expecting the user to search facilitates retrieval. Such an approach is adopted by the second visualization technique, which permits the visualization of large numbers of nodes. An instance of this visualization is shown in Figure 6b. A circular layout is employed to display nodes, organizing them only by time. Recent pages are displayed close to the center of the visualization area, while old pages are displayed close to its borders. As the existence of large numbers of nodes can result in cluttering, a filtering slider is included in the control panel. This slider controls the maximum number of links that are displayed. Low slider values allow only the most relevant links to be shown. Figure 7 illustrates different instances that exhibit how the slider filters the display of links in both visualizations.

The visualizations are not automatically updated when a new page is visited. The user can request an update by clicking on the refresh button which is included in the control panel. This action sends an HTTP request to the page modifier, which responds with new history information. The main limitation of our approach is that long browsing sessions can result in large numbers of nodes, whose maintenance and visualization may not be feasible. We tackle this problem by removing linking information of relatively old page visits.

## 7. CONCLUSIONS AND FUTURE WORK

After reviewing related work, we hypothesise that it is the adaptable combined with the adaptive attributes that will afford a strong win for users involved in information gathering tasks in particular. To that end, in this paper we have presented a novel approach to support link *adaptation* through *adaptable* direct manipulation techniques. The paper proposes how this approach is a potentially effective addition to adaptive hypermedia techniques. As an exemplar of this claim, we have proposed two cases, navigation through pages and assessment of history lists, as applications for this approach, and have presented the model and architecture to support this work.

Our major future priority is the evaluation of the prototype and the deduction of valid conclusions about whether and how our approach can assist the browsing task of a user. Essential questions that need to be answered are: Does the system speed up the process of locating interesting information? How does it compare with purely adaptive systems? How do users interpret the weights on the sliders, and how do they manipulate them to describe their interests? How do the history visualization techniques compare with other existing re-visitation tools such as bookmarks? How is the display of linking information in the history visualizations helpful?

We are also currently investigating adaptation techniques others than adaptive link annotation, which could be incorporated into the same framework [28]. Figure 8 illustrates the application of a fisheye view with multiple focal points to the content of a web page. Paragraphs related to the user interests are on focus, while unrelated paragraphs are distorted but not hidden from the user's view. Similar techniques have been applied to support collaboration and workspace awareness in situations where multiple people view or work on the same document [11]. The above approach suggests that direct manipulation can be applied to control not only links but the whole content of a page, helping the user to locate interesting information within large pages.

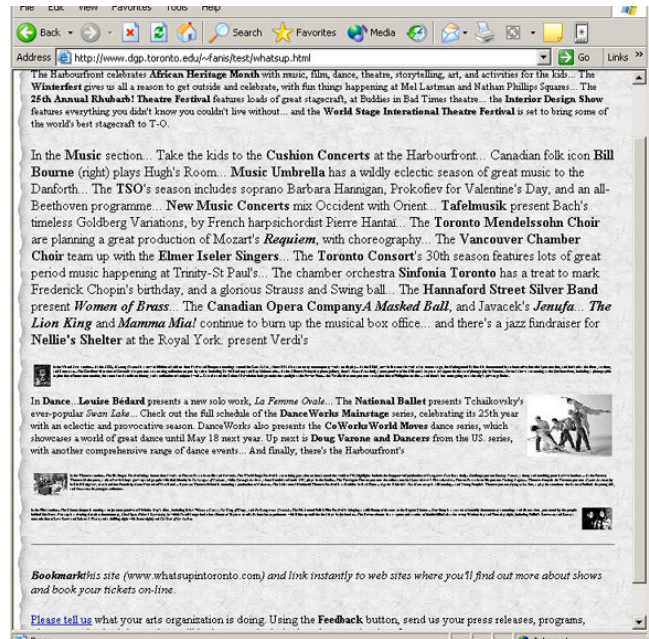


Figure 8. Fisheye view of a Web page reflecting the fact that the user is interested in information about music<sup>1</sup>

Finally, a future goal is to study how direct manipulation could be applied by using automatically generated descriptors of Web content rather than expecting the user to specify interests. For instance, assuming that the user visits the page in Figure 8, the system would automatically suggest a set of sliders for the individual topics on the page: theatre, music, dance, etc. In this way, direct manipulation would be based on the content of individual pages or even the content of groups of pages, for example, the pages of a whole Web site. Although such a scenario seems distant in today's unstructured Web, it is realistic in the world of the Semantic Web, where structure and semantics are supported.

## 8. ACKNOWLEDGMENTS

Thanks to Bowen Hui for helpful comments on earlier versions of the paper.

## 9. REFERENCES

- [1] Abrams, D., R. Baecker and M. Chignell (1998). Information Archiving with Bookmarks: Personal Web Space Construction and Organization. CHI'98 Conference on Human Factors in Computing Systems, Los Angeles CA, United States, ACM Press.
- [2] Ahlberg, C. and B. Shneiderman (1994). Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. Conference on Human Factors in Software, CHI '94, Boston, MA, United States, ACM Press.
- [3] Bauer, T. and D. B. Leake (2001). WordSieve: A Method for Real-Time Context Extraction. 3rd International and Interdisciplinary Conference, Context, Dundee, Scotland, Springer Verlag.

<sup>1</sup> The screenshot is a modified version of a page taken from an online guide about arts and entertainment in Toronto: <http://whatsuptoronto.com>.



- [4] Bederson, B. B. (2000). Fisheye menus. 13th Annual ACM Symposium on User Interface Software and Technology, UIST, San Diego, California, United States, ACM Press.
- [5] Brusilovsky, P. (1996). Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction* 6(2-3): 87-129.
- [6] Cockburn, A. and S. Greenberg (1999). Issues of Page Representation and Organisation in Web Navigation Tools. OZCHI'99 Australian Conference on Human Computer Interaction, Wagga Wagga, Australia.
- [7] Cockburn, A., S. Greenberg, B. McKenzie, M. Jasonsmith and S. Kaasten (1999). WebView: A Graphical Aid for Revisiting Web Pages. OZCHI'99 Australian Conference on Human Computer Interaction, Wagga Wagga, Australia.
- [8] Conklin, J. (1987). Hypertext: an introduction and survey. Computer, IEEE Computer Society Press 20(9): 17-41.
- [9] El-Beltagy, S. R., W. Hall, D. D. Roure and L. Carr (2001). Linking in Context. 12th ACM conference on Hypertext and Hypermedia, Aarhus, Denmark, ACM Press.
- [10] Furnas, G. W. (1986). Generalized Fisheye Views. CHI '86, Boston, Massachusetts, USA, ACM Press.
- [11] Greenberg, S., C. Gutwin and A. Cockburn (1996). Using Distortion-Oriented Displays to Support Workspace Awareness. *People and Computers XI (Proceedings of HCI '96)*, Springer-Verlag.
- [12] Head, M., N. Archer and Y. Yuan (2000). World Wide Web Navigation Aid. *International Journal of Human-Computer Studies* 53(2): 301-330.
- [13] Hirashima, T., N. Matsuda, T. Nomoto and J. Toyoda (1998). Context-sensitive filtering for browsing in hypertext. 3rd International Conference on Intelligent User Interfaces, San Francisco, California, US, ACM Press.
- [14] Hohl, H., H.-D. Boecker and R. Gunzenhaeuser (1996). HYPADAPTER: An Adaptive Hypertext System for Exploratory Learning and Programming. *User Modeling and User-Adapted Interaction* 6(2-3): 131-155.
- [15] Joachims, T., D. Freitag and T. M. Mitchell (1997). WebWatcher: A tour guide for the World Wide Web. *IJCAI*.
- [16] Kaplan, C., J. Fenwick and J. Chen (1993). Adaptive hypertext navigation based on user goals and context. *User Modeling and User-Adapted Interaction* 3: 193-220.
- [17] Kushmerick, N., J. McKee and F. Toolan (2000). Towards Zero-Input Personalization: Referrer-Based Page Prediction. *International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, AH 2000, Trento, Italy*.
- [18] Lieberman, H. (1995). Letizia: An Agent That Assists Web Browsing. 15th International Joint Conference on Artificial Intelligence, IJCAI-95, Montreal, Quebec, Canada, Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.
- [19] Mladenic, D. (1996). Personal WebWatcher: Implementation and Design. Slovenia, Department of Intelligent Systems, J.Stefan Institute.
- [20] Nielsen, J. (1990). Hypertext and Hypermedia, Academic Press.
- [21] Nikunj, D., Z. Quible and K. Wyatt (2000). Cognitive design of home pages: an experimental study of comprehension on the World Wide Web. *Information Processing and Management* 36(4): 607-621.
- [22] Pazzani, M. J., J. Muramatsu and D. Billsus (1996). Syskill & Webert: Identifying Interesting Web Sites. 13th National Conference on Artificial Intelligence, Portland, OR, US.
- [23] Salton, G. (1991). Developments in automatic text retrieval. *Science* 253: 974-979.
- [24] schraefel, m. c. (2000). ConTexts: Adaptable Hypermedia. *Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000, Trento, Italy, Springer-Verlag*.
- [25] Shneiderman, B. and P. Maes (1997). Direct manipulation vs. interface agents. *Interactions, ACM* 4(6): 42-61.
- [26] Stanyer, D. and R. Procter (1999). Improving Web usability with the link lens. 8th International WWW Conference, Toronto, Canada.
- [27] Thüring, M., J. Hannemann and J. M. Haake (1995). Hypermedia and cognition: designing for comprehension. *Communications of the ACM* 38(8): 57-66.
- [28] Tsandilas, T. and m. c. schraefel (2003). Adaptive Presentation Supporting Focus and Context. *Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, AH2003, Nottingham, UK*.  
<http://www.wis.win.tue.nl/ah2003/proceedings/>.
- [29] Weinreich, H. W. R. and W. Lamersdorf (2000). Concepts for Improved Visualization of Web Link Attributes. 9th International World Wide Web Conference on Computer Networks, Amsterdam, The Netherlands.
- [30] Wickens, C. D. (1992). *Engineering Psychology and Human Performance*, Harpers, New York.
- [31] Zellweger, P. T., B.-W. Chang and J. D. Mackinlay (1998). Fluid links for informed and incremental link transitions. *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia: links, objects, time and space-structure in hypermedia systems*, Pittsburgh, Pennsylvania, United States, ACM Press.
- [32] Zhu, E. (1999). Hypermedia Interface Design: The Effects of Number of Links and Granularity of Nodes. *Journal of Educational Multimedia and Hypermedia* 8(3).