An Independent Navigation Model for Consistent and Personalized Web Navigation

Guangzhi Zheng
Georgia State University
gzheng@cis.gsu.edu

Abstract

There are various designs to improve website navigation. However, a fundamental issue is still not addressed in all these designs: the consistency of navigation models across different websites. In addition, almost all of the current designs are static and passive; users have to adapt their navigation behaviors to each model and follow it all the time on a particular website. The problem is not easily solved by current usability guidelines. This research proposes a new Independent Web Navigation model to address these problems. The proposed solution separates and models navigational information independently, provides a consistent and personalizable client interface for the complex and unpredictable web, and offers rich functionalities and flexibility for users to easily manage and adapt web navigation to their needs.

Keywords:

Web navigation, web usability, browser, user interface, web information structure

Introduction

Web browser has become a major type of user interface for information seeking and processing on the Internet. However, as more webpages and information are provided on the web, people actually spend more time on seeking information instead of using it. Jul and Furnas (1997) summarized two generic tactics to seek information on the web: querying and navigation. Querying, or searching, is the process of "submitting a description of the object (for instance, keywords) to a search engine which will return relevant content or information" (Jul and Furnas 1997). Navigation is the action of "moving oneself sequentially around an environment, deciding at each step where to go next based on the task and the parts of the environment seen so far" (Jul and Furnas 1997). Users use these two tactics together to obtain information they want on the web. While searching has drawn more attention for the past few years, navigation is still a fundamental way, and even the "last mile", of getting useful information. For example, users still need to navigate through searching results to evaluate the relevance and usefulness of them.

Navigating in an information overloaded space is not an easy task. People often feel entangled and disoriented when overloaded with massive amount of information on the web, a problem often referred to as "getting lost" (Lazar et al. 2003, Nah and Davis 2002). They tend to lose sense of location, direction and context (Head et al. 2000), especially when there is minimum assistance provided. Users get lost if they are not used to the website design, or if information is buried deep in the site and cannot be located, or if information is isolated without any linkage to related information (information island). Users often reported frustration when "getting lost" (Lazar et al. 2003).

A good navigation mechanism can ease the problem of getting lost and improve the information seeking effectiveness. Navigation is also a key factor of web usability (Palmer 2002), which studies the ease-of-use of web applications and interfaces. Based on user behavior studies, usability guidelines are offered to guide developers to develop better user interfaces. These usability studies and guidelines indeed have a positive impact on web application interfaces, but they still have not effectively solved following problems:

1. Navigation models and designs are inconsistent across different websites (heterogeneity of navigation elements (Hitz et al. 2006)). There are numerous navigation designs which are quite different in terms of structure, location, sequence, style, graphics, animation, etc. These differences create adapting difficulties for normal users (Silver and Ward 2004); they don't like to be "forced to learn a special way of doing things" (Nielson 2002). For one example, users often cannot find sitemaps because of the complexity and inconsistency of homepages (Neilson 2002). Even the best designed navigation cannot address this problem because web navigation design is focus on an individual site thus hardly can be coordinated with other sites. Introducing more navigation designs can only worsen the problem.

2. Most of web navigation designs are static, fixed for all users, all the time. They lack interactivity, are site orientated and consider little about users' different navigation behaviors and information needs. For example, most of these designs offer no personalization or customization on interface and content structure, which is a common expectation for the purpose of ease-of-use (Blom and Monk 2003).

3. Current web navigation designs lack editability and manageability. They do not provide rich functions for personal information management (PIM) (Teevan et al. 2006) such as sorting, indexing, user annotation, organization, categorization and sharing. This actually results from the inherent limitations of HTML based web interface.

Is there a navigation model that can address all the problems above? The objective of this research is to develop a new web navigation model that provides consistent, personalizable and manageable navigation to web users. Based on the proposition to enhance navigation capability of the web browser itself (Head et al. 2000, Nielson 2002), the Independent Web Navigation model is proposed and illustrated in the following sections. The new model supports a wide range of web information structures and user activities on web navigation. A prototype is created to show the merits of this navigation model.

Related work

Web navigation designs

Many of the prior web navigation design projects focused on web usability. Usability studies assess the ease-of-use of user interfaces. Web usability studies extended these basic principles to the web environment as web browser became a major interface for people to deliver and access information and services. Palmer (2002) suggested 5 factors to measure web usability and navigation was one of them.

A major consideration in web navigation design is to avoid getting lost. Getting lost is a common problem in the non-linear information space like web. The theory of human information

processing (Miller 1956) suggests that when navigating, a user tries to construct a structure map of the information space and a navigation path in mind. This information is stored in short term memory which has limited capacity. As the user is reading content, he/she will have difficulty in memorizing that structure. When the structure of information vanishes from the short term memory, he/she tends to get lost. To relieve memory overload and avoid getting lost, web usability studies suggest general guidelines such as (Neilson 2002, Proctor et al. 2002): 1) visualizing the structure of information space - for example, a site map; 2) providing easy and flexible access to the navigation information; 3) providing context cues; 4) providing navigation trace/history.

Based on these guidelines, a number of navigation models were designed and studied. Zhang and Salvendy (2001) proposed Structure Preview Design (dropdown menu that constructs a major navigation map displayed at the top of the page) and reported it had a significant improvement on user performance. Bederson (2000) designed a Fisheye Menu that can provide focus in a long menu. Park and Kim (2000) proposed a similar context menu design which was able to display parent and child menu items in the same webpage. The advantage of this design is the easy access to several levels of content. But in practice the behavior of expandable menus are very inconsistent; and because submenu items are often hidden to save space (especially in horizontal designs), context cues are lost.

Some other designs make the structure information stay in sight all the time for easy access. For example, floating menu is at a fixed position on the screen as users scroll up and down the page (twinhelix.com). Frames are also used to make structures stay in sight all the time. Hochheiser and Shneiderman's (2000) simultaneous menu makes use of frame for different levels of menus. However, frames have been criticized for low usability that it adds confusion and has the difficulty of bookmarking and printing (Nielson 1996).

Research on using Windows GUI in web applications

Most navigation designs are directly embedded in HTML web pages. There are a number of problems of this approach, especially when websites are structural complex and content oriented. First, it is slow and there is significant delay and interruption between webpage transitions. Second, styling techniques and complex scripting are often used to boost aesthetics and utility, but their incompatibility frequently creates confusions when they don't behave as what users expect. Third, it is difficult to interact with local resources. These problems are mainly caused by the limitations of HTML webpage interface. Recent development on AJAX (Asynchronous JavaScript and XML) begins to address these problems and offers more UI responsiveness and functionalities. But these script based solutions still lack consistency.

Another approach is to improve the navigation capability of web browser itself (Head et al. 2000). Compared to HTML web interface, Windows desktop graphical user interface (GUI) has some unique advantages: it has a richer set of UI controls; it is highly interactive and responsive; it is easy to interact with local resources; and it is highly consistent in appearance and behavior. However, current browsers have very limited capability to support these advantages for web navigation (Head et al. 2000, Lazar et al. 2003, Silver and Ward 2004). For example, browsers usually have back and forward buttons to assist navigation, but these two buttons often interfere with a website's own navigation system. Browsers also provide bookmark function, which is useful for quick access but lacks context cues.

There had been some studies and practices exploring the use of windows components for web applications. Head et al. (2002) designed a tool "MEMOS", which is displayed in a separate window, to improve the use of navigation history. Pitkow et al. (2002)'s OutRide used a similar browser sidebar for personalized searching. Lin (2004)'s Personalized Indexing and Quan et al. (2003)'s multiple categorization both utilized windows GUI component to index and categorize web pages. Lycos (http://www.lycos.com/) uses "side search", a browser component, to display searching results. Although these tools do not deliberately utilize windows GUI advantages for web applications, they did show some promising usage and they can be generalized to a broader range of web applications, including navigation.

An Independent Web Navigation Model

Neilson (2002) suggests that "better awareness of the user's navigation probably requires some amount of integration with the web browser". In accordance with Nielson's notion, this paper proposes "Independent Web Navigation", in which navigational information (web content structures) is separated from other web content and is processed and managed by a client component in web browser. The model has three most important elements (Figure 1). First, navigation information (content structures) is modeled and managed independently. This abstraction provides the basis of processing consistency. Second, navigation information is processed consistently by a web browser component. This component offers rich visualization and management functionalities for users. Third, a specific protocol is established to discover, load, transform and share such information. The following sections will describe these elements in detail.

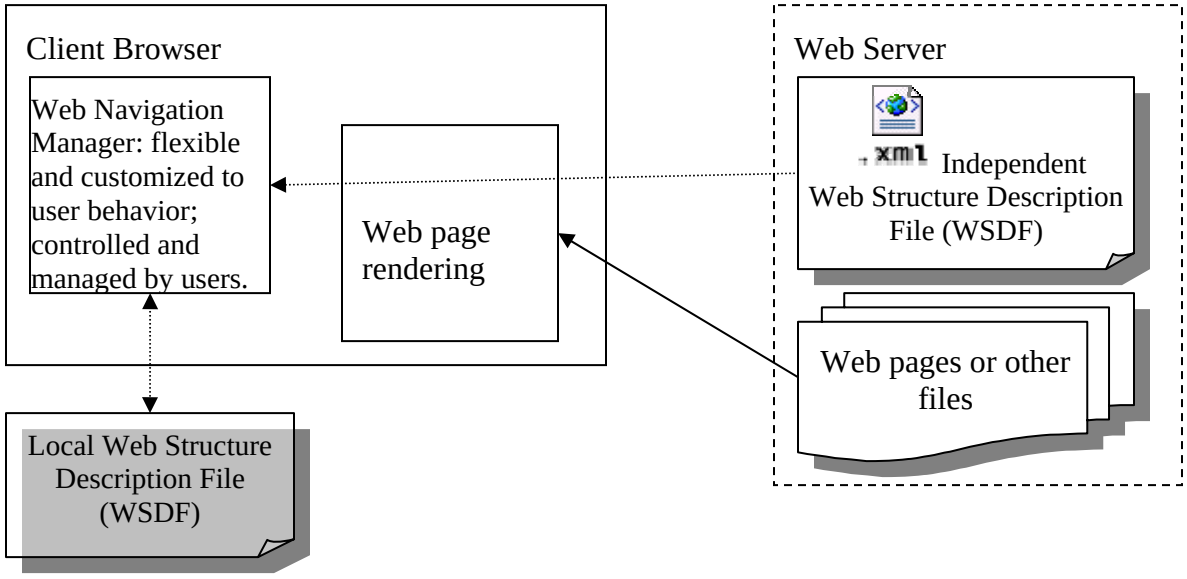

Figure 1: Independent Web Navigation Model

Web Structure Description File (WSDF)

WSDF is an XML based file to model and record a wide range of web content structures. XML is a semi-structured way to describe information and it is natural to use XML to organize structure information for web navigation (Proctor et al. 2002). XML is also a text-based open standard, which makes it perfect to be delivered via the web and processed by various applications easily. Table 1 summarizes a number of XML-based specifications used commonly to organize and publish information.

Specifications	Description
Outline Processor Markup Language (OPML) by Dave Winer (http://www.opml.org)	OPML allows organization and exchange of outline-structured information
XML Bookmark Exchange Language (XBEL) by the Python XML-SIG (http://pyxml.sourceforge.net/topics/xbel/)	XBEL is a format for the interchange of bookmark data.
Really Simple Syndication (RSS 2.0) by Dave Winer (http://blogs.law.harvard.edu/tech/rss)	RSS is a widely implemented format to publish news items and other dynamic information feeds.
ASP.Net 2.0 site map (http://msdn2.microsoft.com/en-us/library/aa479338.aspx)	An ASP.net 2.0 framework specific web site structure file. It is simple but almost limits its use in ASP.Net environment. For example, it is not publicly accessible; each item needs to have a unique URL.
Google sitemap (http://www.sitemaps.org/)	A file prepared to inform search engines about pages on their sites that are available for crawling.

Table 1: XML-Based Specifications to Publish Web Content

These existing formats can be used to record a wide range of structures because of their similarities to organize hierarchical information. They are options to construct WSDFs. However, each of them was developed to represent only a certain perspective of web content. They may not be completely suitable for all web content structures and general navigational activities. For example, most of them assume a hierarchical structure, and they do not support multi-categorization. WSDF may still need to have its own format to accommodate other kinds of information structures and navigation patterns.

Here is an example of a preliminary WSDF document, followed by explanations of major elements and attributes.

```
<WSDF version="1.0">
<Head>
        <StructureName>AIS World Sitemap</StructureName>
        <StructureFormat>Hierarchy</StructureFormat>
        <metadata name="WebRoot">http://isworld.org</metadata>
        <metadata name="Publisher">AIS</metadata>
        <metadata name="FileEditor">AIS Web Master</metadata>
        <metadata name="UpdateDateTime">12/08/2006 10:20:22</metadata>
</Head>
```

```
<Section title="AIS World" url="http://isworld.org/" type="absolute" baseUrl="http://isworld.org/">
    <Link url="http://aisel.isworld.org/" title="e-Library" type="absolute"/>
    <Link url="http://www.isfacdir.org/" title="Faculty Directory" type="absolute"/>
    <Link url="http://jais.isworld.org/" title="Journal of AIS" type="absolute"/>

    <Section title="Today on AISWorld" description="Messages from the AISWorld distribution list."
    baseUrl="http://lyris.isworld.org/lib/bodyout.pl?messageno=">
        <Link url="301698" title="EU/IST-Africa 2007" type="relative" description="CFP"/>
    </Section>
</Section>
<WSDF>
```

The format begins with a “Head” section to provide document level information, such as name, date, editor, etc. The “StructureFormat” element specifies the information organization model of the document: “Hierarchy” means information is hierarchically structured; other options are “Grid” (or “Table”, where information can be presented in a 2 dimensional view) and “Multi-category” (content items are defined by keywords, or predefined categories, rather than hierarchies). “Metadata” tags in the “Head” section are optional to include more information for other intended use.

The body of the XML document consists of one or more “Section” tags, depending on document’s structure format and editor’s choice. A “Section” is logical group (can be nested to represent hierarchy) of web content sources. It may or may not link to a certain web source (“url” attribute is optional). Sections are also non-exclusive. “Link” tag is the leaf level node that provides “url” to a web source; links are always wrapped in sections. The “url” attribute can be “relative” or “absolute”, depending on its relevance to peer links in the same section.

Windows GUI based web navigation manager

Web navigation manager is an integrated component of web browser on the client computer. It is part of the browser (integrated) and is basically of a windows GUI. It is independent to any website and any navigation pattern, thus it offers a “consistent” interface and navigation model to a particular user. The major functionality of this client is to visualize and manage WSDF (potentially with multiple formats supported). Because navigation information is organized in such a common understandable format, the client can provide a consistent and user preferred view even the information is coming from different sources. This client is also easily and consistently accessible, with minimum mouse or keyword actions.

The client is highly personalizable in terms of location, view style and appearance. The location of the manager can be customized to various screen positions (top, bottom, left, right, inside or side-by-side to browser, docked and float) using various GUI controls (side bar, panel, window, menu bar, tool bar, gadgets). The visualization can be customized to various view styles, such as hierarchical tree view, HTML view, tab view, grid view, and even advanced visualizations such as map view, 3D view, etc. The look and feel of these views can also be personalized through themes or style sheets: color, icon, size, shape, font, animation, etc. View styles and appearance themes are easily expandable to included third party designs.

One of the novelties of web navigation manager is, as the name subjects, it is not just provide a view of information, but also satisfies personal information management needs (Teevan et al. 2006). The navigation manager offers rich management functions and makes navigation more

interactive and user-controlled. This is where it offers the greatest potential, because current navigation designs almost ignore such need and offer no such functions. Examples of these management functions include editing, annotation, organization, importing/exporting, composition, searching, indexing, sorting, filtering, zooming, etc. Because the client is of desktop GUI, it is inherently faster and more responsive, and supports complex interactions like drag-and-drop, double-click, double-point, shortcut keys, etc.

Dynamic WSDF discovery and loading

To provide seamless access and use of these structures, WSDF should be loaded dynamically and automatically. An ideal way of loading is to automatically load the corresponding WSDF when a certain site is being visited. To achieve this goal, a standard protocol needs to be established. First, WSDF should be placed in a specific location on the web server for dynamic discovery (for example, the root directory). It can be accessed by browsers and other web applications through standard HTTP request. One website (or one host) can have one WSDF describing the complete website (host); or it can have multiple WSDFs with each describing a sub-site or a major section of information. In the latter model, the WSDF under the domain or host root becomes a "directory" or WSDFs on the same site (host). Second, when a user requests a URL, the Web Navigation Manger will analyze the URL and try to locate the WSDF in the root directory. It will analyze the document and find the most matching sub-level WSDF to the current URL and load it. Third, as users navigate the site in browser, the manager will dynamically monitor user's request and always give location indication (for example, highlighted item) in WSDF view so users will always have a sense of location. Last, users can always manually "zoom in" or "zoom out" in these WSDFs.

WSDF can also be created and stored locally. In this case, all local WSDFs will be stored in a specific local directory and the Web Navigation Manger can easily load them on user's choice.

Prototype

An initial prototype, ZWebNav, is developed to experiment the new web navigation model. Several sample WSDFs are created to model some typical websites, representing common web content structures of sitemap, product catalog, directory, and book table of contents. The prototype is developed based on .Net framework 2.0 Windows environment. The client manager component is designed as part of the browser. Figure 2 shows a screenshot of the prototype loaded with the "isworld.org" WSDF. Different sources of contents can be loaded to the same view, displayed and used consistently.

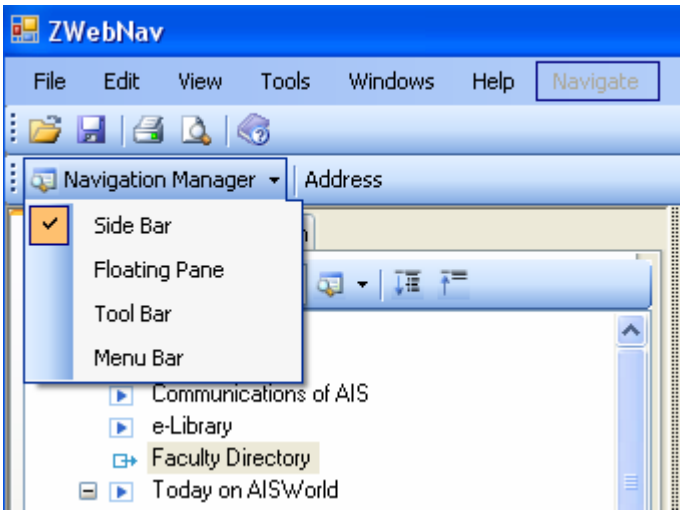

Figure 2: Navigation Manager at Various Positions

The navigation manager can be flexibly activated and hid by select/deselect an item in the "Navigation Manager" dropdown menu (Figure 2). There are

four positions designed to place navigation manager: side bar, floating panel, toolbar and menu bar. Side bar is located (docked) at the left side of the browser; its width can be easily adjusted (Figure 3). Floating panel is a separate layer of window that can be placed anywhere on the screen (Figure 4). Its size can be adjusted freely and can be maximized to full-screen. Tool bar is located at the top of the browser as a set of icon buttons and text labels. Menu bar is located in the standard Windows menu bar as dropdown menus.

Two view styles are developed with the prototype: HTML view and Windows tree view. HTML view provides a web-like style. It utilizes XSLT to transform XML-based WSDF into standard HTML, and uses CSS style sheets (user made or website provided) to personalize the look and feel (Figure 3). When a user clicks on a URL, the requested web page will be displayed and the URL will be highlighted.

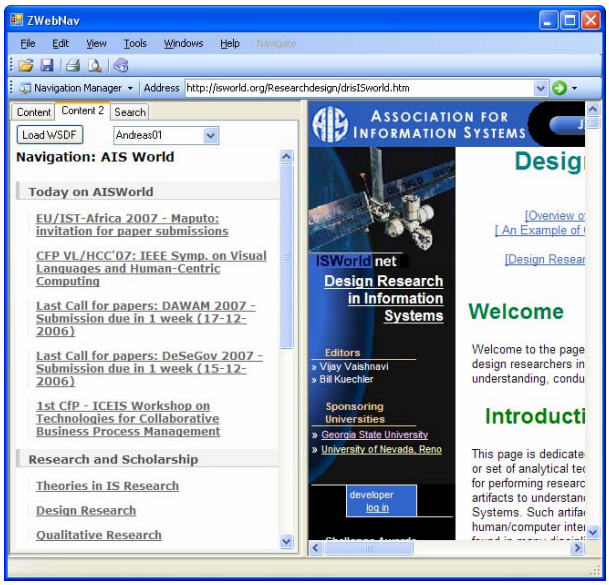

Figure 3: HTML View in Side Panel

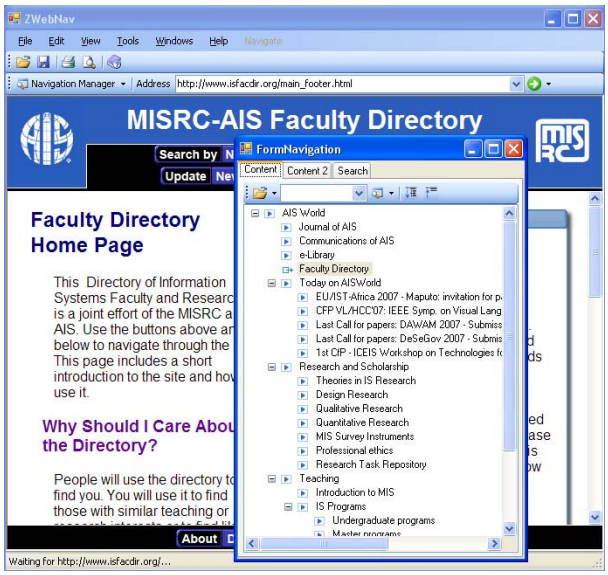

Figure 4: Tree View in Floating Panel

Windows Tree view control is a common Windows GUI control to hold hierarchical content (Figure 4). It supports standard tree view functions like expanding/collapsing, forward/backward, going up-level/down-level. When a user clicks on an item, the requested web page will be displayed, and the current active item is highlighted and also indicated by a different icon. This explicit context cue will give users clear sense of orientation and ease their memory load.

The navigation manager also provides a search function. This function allows a user to do a search based on a loaded WSDF; the search results will be displayed with the original hierarchy preserved (Figure 5). Such searching against condensed structure information will be more effective than full site search is.

Discussion and evaluation

Preliminary evaluations during the prototyping process already showed user's comfort and preference to use this new model of web navigation. A more formal evaluation of the model is currently under preparation. This section provides a detailed conceptual analysis of the new navigation model.

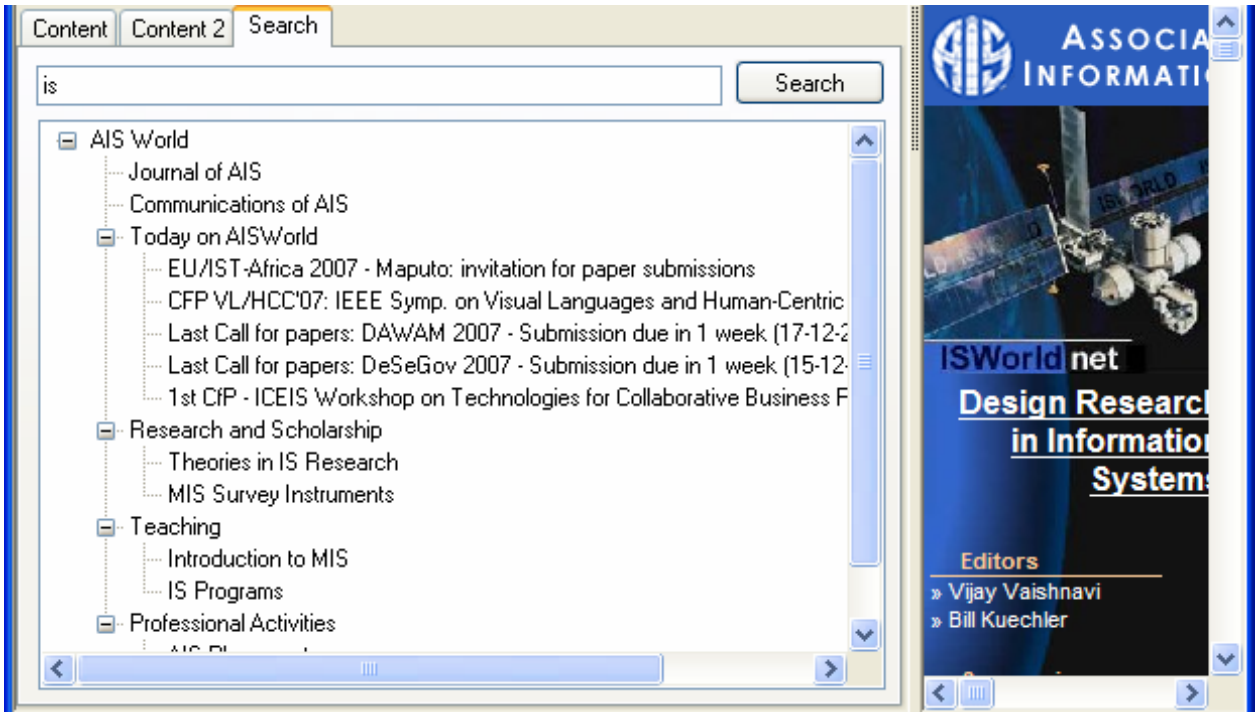

Figure 5: Search against WSDF

There two unique features of the Independent Web Navigation model: 1) the navigational information is separately modeled and organized using XML files; 2) a GUI based component on the client computer is used to display and manage navigation information. These features nicely preserve and expand some of the important good-practices from web navigation usability studies:

1) *Visualizing the structure of information space*. The visualization provides an overview and highly condensed information for quick browsing and easy locating of content. The navigation manager can provide this visualization as good as (and better than) any current HTML designs, because the HTML view (one of the many view styles) basically utilizes the same technologies as all other websites do.

2) *Providing easy and flexible access to the structure*. The client component can be easily activated and hid, offering a clear separately presentation of navigation information and main content. It also can always stay in sight when a user needs it. This helps a user's attention easily move between structure and content, without excessive interactions.

3) *Providing context cues*. Context cues are indications of location and relationships to other contents. The navigation manager provides these clear context cues dynamically and automatically within the visualization.

These unique features of this model also bring unique merits that address the problem listed in the introduction section:

1) *Navigation consistency*. Consistency is one of the important aspects of usability in user interfaces (Nielson 1999). The separated modeling and processing of navigation information provide the basis for navigation consistency. The XML based WSDF is flexible and extensible to support a wide range of content and format; all structures of these content can be displayed and used in a consistent interaction model, independent from providers' own designs.

2) *Personalization and customization*. The web navigation manager is client based, so it is easy to offer flexible and easy customization and personalization. Personalization in navigation is largely missing in current website designs, although it is commonly expected and welcomed by users today. A personalized navigation can give users comfortable and enjoyable way of web information seeking.

3) *Rich functionality for personal information management (PIM) need*. The web navigation manager provides unprecedented potential for users to manage their navigation. GUI based interface offers the responsiveness and richness for these functionalities, and XML based information format provides openness to deliver, share, synchronize and integrate content. The richness of PIM operations will enhance user experience. This is definitely an opportunity for future research.

Conclusion

Web navigation has suffered the problem of inconsistency, static and passiveness. To address these issues, a new design perspective is needed to go beyond web page design and usability guidelines. This paper proposed a consistent and adaptive navigation model for the complex and unpredicted web environment. It will alleviate users' learning frustration and improve their information seeking effectiveness.

The proposed model offers new perspectives to design web navigation. First, at the heart of this model is the Web Navigation Manager, a Windows GUI integrated in browser that processes XML based website structure information. Using browser itself to facilitate navigation has several merits over traditional navigation designs. The research is an early attempt to explore this use of hybrid interface for web applications. A hybrid interface refers to a user interface with both windows GUI element and HTML web element. Each element has its own advantages and disadvantages. This research contributes to an initial understanding of this type of interface.

Second, the structure information is separately and explicitly represented and organized in XML. This allows easy reuse of structural information. They can be created, extracted, edited, combined, exchanged and formatted easily and universally. The work load now can shift more on organizing and providing website structure information. This will facilitate the delivery of more useful and high quality information. Future research will also focus on a more complete and richer version of WSDF and an enhanced protocol for WSDF discovery and loading.

Reference

1. Bederson, B. (2000), "Fisheye Menus", Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST), November 5-8, 2000, San Diego, California, pp. 217-225
2. Blom, J. O. and A. F. Monk (2003), "Theory of Personalization of Appearance: Why Users Personalize Their PCs and Mobile Phones", Human-Computer Interaction 18: 193-228

3. Head, M., N. Archer, Yuan (2000), "World Wide Web Navigation Aid", International Journal of Human-Computer Studies 53(2): 301-330
4. Hits, Martin, Gerhard Leitner, Rudolf Melcher (2006) “Usability of Web Applications”, in “Web Engineering”, edited by Gerti Kappel et al., John Wiley and Sons, 2006
5. Hochheiser, H. and B. Shneiderman (2000), "Performance Benefits of Simultaneous Over Sequential Menus as Task Complexity Increases", International Journal of Human-Computer Interaction 12(2): 173-192
6. Jul, S. and G. W. Furnas (1997), "Navigation in Electronic Worlds: A CHI 97 Workshop Report", SIGCHI Bulletin 29(4)
7. Lazar, J., K. Bessiere, et al. (2003), "Help! I'm Lost: User Frustration in Web Navigation", IT&Society 1(3): 18-26
8. Lin, Shin-jeng (2004), "Enhancing Personalized Indexing with XML", the Tenth Americas Conference on Information Systems, New York.
9. Miller, G.A. (1956) “The magical number seven, plus or minus two: Some limits on our capacity for processing information”, Psychological Review, 63: 81-97
10. Nah, F. F-H. and S. Davis (2002), "HCI Research Issues in Electronic Commerce", Journal of Electronic Commerce Research 3(3): 98-113
11. Neuwirth, C. M. and S. H. Regli (2002), "Usability and the Web", IEEE Internet Computing 6(2): 44-45
12. Nielsen, J. (1989), “Coordinating User Interfaces for Consistency”, Academic Press, 1989
13. Nielsen, J. (1996), "Why Frames Suck (Most of the Time)", Jakob Nielsen's Alertbox, [http://www.useit.com/alertbox/9612.html] Accessed on November 20/2004
14. Nielsen, J. (2002), "Site Map Usability", Jakob Nielsen's Alertbox, [http://www.useit.com/alertbox/20020106.html] Accessed on 10/20/2004
15. Palmer, J. W. (2002), "Web Site Usability, Design, and Performance Metrics", Information Systems Research 13(2): 151-167
16. Park, J. and J. Kim (2000), "Contextual Navigation Aids for Two World Wide Web Systems", International Journal of Human-Computer Interaction 12(2): 193-217
17. Pitknow, J., H. Schütze, et al. (2002), "Personlized Search", Communications of the ACM 45(9): 50-55
18. Proctor, R. W., et al. (2002), "Content Preparation and Management for Web Design: Eliciting, Structuring, Searching, and Displaying Information", International Journal of Human-Computer Interaction 14(1): 25-92
19. Quan, Dennis, Karun Bakshi, David Huynh and David R. Karger (2003), “User Interfaces for Supporting Multiple Categorization”, Proceedings of INTERACT 2003
20. Silver, M. S. and S. G. Ward (2004), "Browser-Based Applications: Positive or Negative Transference?" the Tenth Americas Conference on Information Systems, New York.
21. Teevan, Jaime, William Jones and Benjamin B. Bederson (2006), “Personal Information Management”, Communications of the ACM, 49(1): 40-43
22. Zhang, H. and G. Salvendy (2001), "The Implications of Visualization Ability and Structure Preview Design for Web Information Search Tasks", International Journal of Human-Computer Interaction 13(1): 75-95