A COMPARISON OF HIERARCHICAL AND LINEAR BROWSER BACK MENUS

Daniel Orner and I. Scott MacKenzie
Department of Computer Science and Engineering, York University
4700 Keele St., Toronto, ON Canada M3J 1P3
{orner, mack}@cs.yorku.ca

ABSTRACT

The most common method to revisit web pages is the Back button; however, due to its stack-based nature, pages frequently "pop off" the stack, resulting in those pages being inaccessible from the Back menu. A user study was performed on three alternatives to the regular Back menu – a recency list algorithm, and Histree and FamilyTree, two hierarchical algorithms. Results indicate that both FamilyTree and Recency have strengths and weaknesses. A combination of these two interfaces is suggested to incorporate some of their strengths.

KEYWORDS

Browser history, back menu, hierarchical menus.

1. INTRODUCTION

Studies (Tauscher and Greenberg, 1997; Cockburn and McKenzie, 2001) have shown that the majority of web pages visited by users have been recently seen by them. Revisitation is therefore an important part of web browsing. The majority of revisitation tasks are accomplished via the Back button. (Catledge and Pitkow, 1995; Cockburn and McKenzie, 2001) Studies (Cockburn et al., 2002) have shown that the Back menu is much more efficient than the Back button when used competently.

The main problem with both the Back button and its attached menu is its stack-based behavior (Cockburn and Jones, 1996). To illustrate this problem, let us take a user who starts at page A and visits page B. He then clicks the Back button to return to page A, then visits page C. This navigation path can be shown as A → B ⇦ A → C. The contents of the Back menu will now be {A,C}. Page B has been pruned out of the menu entirely.

2. PREVIOUS WORK

Many solutions to this problem have been proposed. The major issue with nearly all of these solutions (Ayers and Stasko, 1995; Doemel, 1995; Frecon and Smith, 1998; Hightower et al., 1998; Robertson et al., 1998; Gandhi et al., 2000; Milic-Frayling et al., 2003; Milic-Frayling et al., 2004; Jhaveri, 2004) is that they require a separate visible window. This either significantly reduces the screen real estate, or requires the user to continually pop up a new window and close it. Both effects are a considerable detriment to the web browsing activity.

Some work exists on replacing the current stack-based behavior of the Back button with a recency-list implementation (possibly with some variations), which lists the most recently visited pages (Greenberg and Cockburn, 1999; Greenberg et al., 2000; Cockburn et al., 2002; Cockburn et al., 2003). However, several problems with this behavior have surfaced in these same studies. For example, hub-and-spoke behavior (where the user visits several pages from the same "hub" page) often causes problems for recency lists. Either the hub drifts further and further away from the last child page (by placing intervening children between the hub and the current page) or the intervening children begin to drift further and further away from the hub (by placing them *before* the hub). In addition, the studies mentioned concentrate on the Back button rather than the Back menu.

An intriguing study (Wen, 2003) suggests that providing users with navigational context – i.e., where they went and how they got there – could be a useful tool, perhaps replacing thumbnails in giving more modes of recognition. If the user is presented with titles or URLs which are not useful (as is so often the case in web browsing), simply providing cues to the user's previous navigations could allow him to more easily find the desired page. Such information needn't take up precious screen real estate, and is easily integrated into the browser.

3. HIERARCHICAL MENUS

In a separate paper (Orner and MacKenzie, 2006), we present two algorithms which display a hierarchical back menu interface. Both are based on the regular stack interface, but introduce submenus allowing the user to visit pages normally inaccessible from the regular back menu. *Histree*, the first interface, displays the user's full navigational tree by placing pages normally "popped off" the stack into a submenu. The submenu may have more children and submenus (indicating further paths). An example of the behavior of Histree is shown in the following visitation track:

1 → 2 → 3 → 4 → 5 ⇦ 4 ⇦ 3 ⇦ 2 → 6 → 7 → 8

The Back menu as created by Histree would be as follows (the user is viewing Page 8):

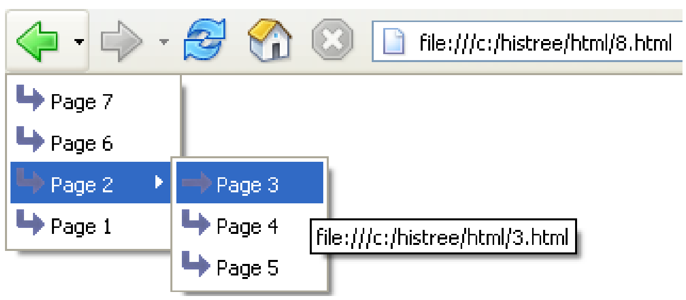

Figure 1. An example of Histree's generated Back menu

FamilyTree, the second algorithm, is similar to Histree, but treats the user's navigation as a graph rather than a tree. Each menu item contains a submenu showing both its children *and* its parents. (Submenus are trimmed down by removing items already displayed in the immediate parent menu.) This has the advantage of treating each page as only a single item, as opposed to Histree which has multiple entries of the same page. This allows the user to easily see *how I got here* as well as *where I went from here*.

For example, the path 1 → 2 → 3 → 4 → ⇦ 3 ⇦ 2 → 4 would create the following menu:

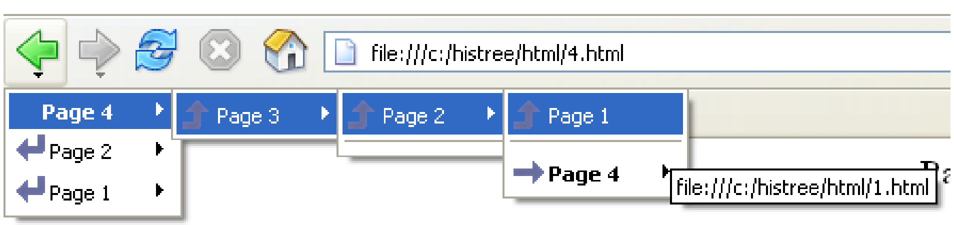

Figure 2. An example of FamilyTree's generated Back menu

4. USER STUDY METHOD

Our user study extended over two months. The goal was to test the efficiency, usability, and comfort level of four back menu interfaces. The experiment was a 4 x 4 x 4 x 10 design, where the Interface and Group variables were between-subjects, while Website and Task variables were within-subjects.

4.1 Participants

Thirty-two paid participants took part in the study. Ages ranged from 16 to 32, with an median age of 22. 19 were male and 13 were female. Participants had between 2 and 14 years of experience using the Internet, with a median of 8 years. All were students at the local university. The experiment took from 30 to 60 minutes.

4.2 Interfaces

Four interfaces were studied: The regular Back menu, Histree, FamilyTree, and a Recency implementation. All were altered to allow no limitation on the number of items displayed (even for the Regular back menu, which by default only displays ten items). This caused the Recency menu, which includes every page visited on a single menu level, to be so large that the menu often required scrolling to reach the older items.

The recency algorithm we used is "Recency with Temporal Ordering Enhancement" presented by Cockburn et al. (1999). This allows the user to navigate back and forth through the recency list without changing it, but when a new link is clicked, the list is reordered to reflect the exact order of visitation.

4.3 Tasks

Revisitation tasks were divided into ten different categories, not including clicking the link in the first place. However, the tasks differ slightly due to context and website (see next section).

1: Revisiting a parent page.
2: Revisiting a parent page with intervening children.
3: Revisiting an ancestor at least three pages up.
4: Revisiting an ancestor at least three pages up with intervening children.
5: Revisiting a sibling page.
6: Revisiting a sibling page one level below the current page.
7: Revisiting a sibling page one level above the current page.
8: Revisiting a past parent (e.g., going from page A to B, then later revisiting page A from B)
9: Revisiting the child after returning from task 8.
10: Revisiting a page from a previous site.

4.4 Websites

Four different websites were used so the users did not see the same pages four times. A small subset of the pages from each website were downloaded to the test computer and altered to facilitate the study: all relevant links were changed to point to local files, and some of the navigation links were removed to ensure that the subject used the back menu rather than the links. In addition, two of the websites had their titles altered to remove useful information, and two had their URLs altered to become unreadable, to test how these common conditions affect the back menu. Note that the titles are immediately visible when the menu is opened, whereas URLs are only viewable via a tooltip.

The four websites were:

1. Hewlett-Packard (HP): Bad titles, bad URLs
2. Amazon.com: Good titles, bad URLs
3. York's CSE department: Bad titles, good URLs
4. City of Toronto site: Good titles, good URLs

4.5 Groups

The users were divided into four groups which rotated the four websites in a balanced Latin Square formation.

4.6 Procedure

The user study made use of the Histree extension for Mozilla Firefox. The extension was altered to pop up a "task window" which presented a task – either a link to click, or a request for the user to use the Back or Forward menus to access a previously visited page. When each task was completed, a button blinked on the status bar inviting the user to click it and present the next task. The Back and Forward buttons were converted to menu buttons, so that every revisitation task required opening up the menu rather than simply clicking the button. The task window remained in the background and could be reopened at any time by clicking the "Tasks" window in the taskbar. All functions besides the back menu (e.g., URL bar, menu bar) were disabled, as were all links pointing to non-local URLs.

The design of the tasks proved challenging. We wanted to measure specific tasks, so those tasks had to be there in a fairly rigid fashion. However, we also wanted the user to feel comfortable browsing; clicking mindlessly is not only unlike a real browsing session, it negates the advantages that the hierarchical menus could offer. Our solution was to present the tasks in a logical way, and to clue the user into the "thoughts" of a theoretical person who follows the paths dictated by the tasks. This would keep the users interested while still making sure their actions adhere to our experiment design. One result of this was that, in an effort to keep the paths natural, each website actually had different numbers of tasks. Also, some tasks were featured in one website but not another, in cases where we could not find a natural way to work them in.

Before beginning, a small demonstration of the interface was given to the user along with an explanation of its workings, and he was given time to familiarize himself with the interface using a set of 10 simple interlinked pages. The user was told to consider this a regular browsing session, to pretend to be the person doing the browsing and to pay attention to the pages viewed, though not necessarily to memorize them. He was told that any task that asked the user to use the Back/Forward menu could be completed using *only* the Back/Forward menu (with the exception of the Regular interface, where this is not true and therefore was not mentioned). He was told that some confusion would inevitably be created, especially in sites which had neither useful titles nor names, and to do the task as best he could, no matter how long it took. The user was asked not to use the keyboard, only the mouse.

4.7 Recorded Data

The following data (dependent variables) were recorded:

- Total time for each task, from the time the "Start Task" button is clicked (or the main window is focused) until the desired page is reached
- Number of pages visited during the task
- Number of menu items hovered over during the task

We feel that time, while a useful indication in a general sense, may not be as important as it seems. This is because many users use their time differently; some continually popped up the task window to reaffirm the task; some scanned the web pages more intently; some simply moved slower than others. In retrospect, it would have been more useful to record the total time that the back menu was open; however, we came to this realization too late to incorporate it into the experiment. We still feel that time can be a useful insight, but not as important as some of the other variables recorded.

We feel that the number of pages visited is the most interesting variable, as the fewer the pages, the more useful the menu; less useful menus will require the user to continually view pages to affirm his location. The number of items hovered over is also noteworthy, as it indicates the movement of the mouse, and how many pages were considered before selecting one.

In recording the data, we gave the users 60 seconds to reach their target. If the task took longer, the user was considered to have gotten "lost", and the value was omitted in the analysis. We later analyse the number of times that users have gotten "lost" for a given interface.

It should be noted that many of the graphs below show large error bars. This occured because some users simply "got it" in a given task, or made a lucky guess, while others became confused and had to work harder to find it. The successful users often reached their target in a few seconds (or one page view), whereas others required almost a full minute and many page views.

5. RESULTS

Results are given as follows for each dependent variable in the study. The second F parameter is the total number of tasks being analyzed (out of 1602); this is lessened when analyzing only a single task, website etc.

5.1 Time

Other than the fact that Regular was faster than the others ($F_{3,1602} = 5.209$, $p < .005$), no other significant total results were detected. However, when analyzing by Website, we found that Histree was significantly slower in Website 4 ($F_{3,332} = 4.028$, $p < .01$), where there were bad titles, but good URLs, than the other three interfaces.

Analyzing by task yielded interesting results. Histree was consistently either slower or about the same as the other interfaces. The exceptions were Task 4 ($F_{3,148} = 9.601$, $p < .0001$), where Recency was slower than all others, and Task 7 ($F_{3,111} = 3.988$, $p < .01$), where FamilyTree was the slowest.

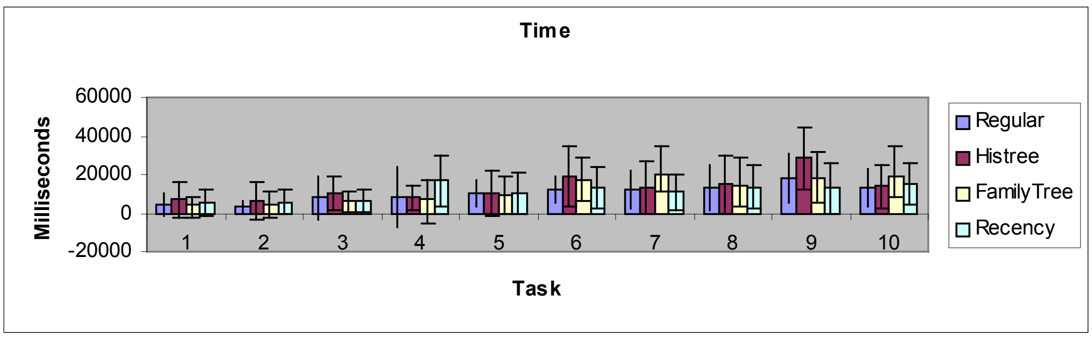

Figure 3. Mean time measured across task and interface, with +1/-1 standard deviation error bars

5.2 Number of Pages

The interfaces differed markedly in the number of pages required to reach the target ($F_{3,1602} = 19.865$, $p < .0001$). From least to most, the order was Histree/FamilyTree, Recency, and Regular. (There was no significant difference between Histree and FamilyTree, but all other relationships were significant.) Analyzing across Website and Task yielded no important discrepancies with this result.

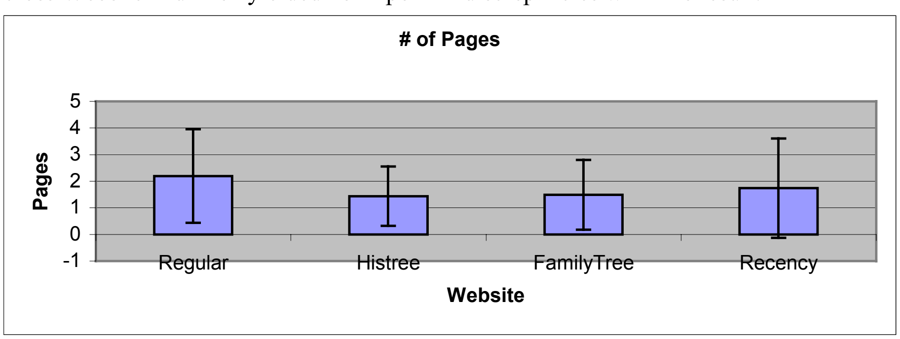

Figure 4. Mean number of pages as measured across interface, with +1/-1 standard deviation error bars

5.3 Number of Hovered Items

Regular showed significantly fewer items hovered over than the other three interfaces ($F_{3,1602}$ = 29.533, p < .0001), which is sensible, given the fact that, using this interface, many pages visited did not require opening the menu at all. Histree/FamilyTree also required fewer items than Recency. An exception was found when analyzing by Task; Recency used less items than FamilyTree for Task 6 ($F_{3,112}$ = 12.688, p < .0001), and it also used less than either Histree or FamilyTree in Task 7 ($F_{3,111}$ = 11.788, p < .0001).

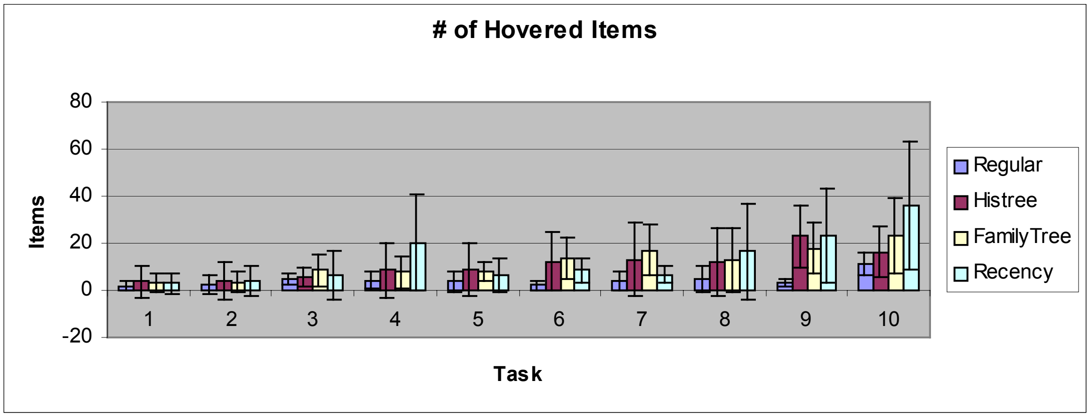

Figure 5. Mean number of hovered items as measured across task and interface, with +1/-1 standard deviation error bars

5.4 Frequency of Becoming "Lost"

A simple comparison indicates that Recency had the fewest instances of tasks which took longer than a minute to complete, followed by Regular, FamilyTree, and Histree. Analyzing across Task, the only exception we found was that FamilyTree did worse than the others in Task 8 ($F_{3,92}$ = 3.286, p < .05). No interesting results were discovered when analyzing across Website.

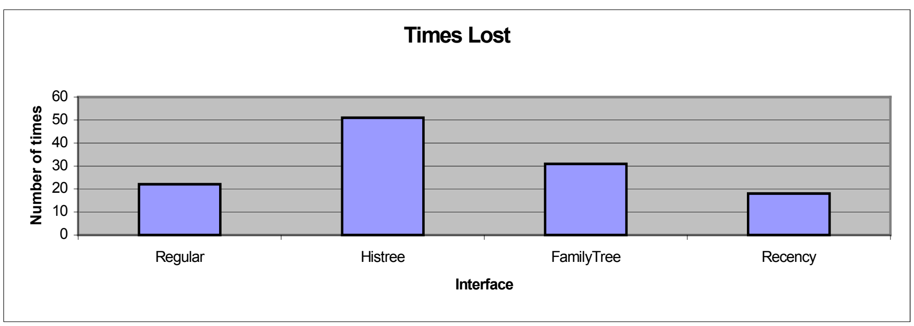

Figure 6. Total number of times "getting lost" for each interface.

5.4 Grouping

We had divided the users into four groups (with different orders of websites) in an attempt to compensate for learning effects. We did find a significant Group effect for the number of page views ($F_{3,1602}$ = 3.646, p < .05) and number of instances of being "lost" ($F_{3,1602}$ = 4.600, p < .005). However, we do not feel that this compromises the overall analyses and results.

5.5 Questionnaires

We asked each user to fill out a questionnaire, where they were asked five questions on a five-point Lickert scale:

- **How easy was it to find the pages you were looking for using the back menu?**
- **How "lost" were you? Were you confident in knowing where you were and where you'd been?**
- **Were you confused about how the back menu worked? Were you confident in understanding it?**
- **How easy was the back menu to use in general?**
- **Would you use this back menu every day?**

Unfortunately, not a single question proved statistically significant across any interface. Perhaps we could have obtained better questionnaire results had we given the user more than one interface to try out, but we wanted to test the user returning to previous sites, and we also did not want to have the user execute the same tasks on the same sites multiple times. It is interesting to note that despite Histree's fairly uniform bad results, users did not appreciably feel that it was less helpful or useful than the other interfaces.

6. DISCUSSION

Before beginning, it is important to note the difference between the Regular interface and the other three. Because of the "pop-off" problem which is the entire point of developing alternative solutions, links could *not* be disabled during revistation tasks as they were in the other interfaces. This means that users had an extra degree of freedom in reaching the desired page, as they could simply click a link they saw, whereas the other interfaces required usage of the menu at all times. We wanted to focus on the usability of the menus, so we did not want to enable links for the other interfaces. However, this influenced the data of the Regular interface. We believe that by using the other three interfaces in a "natural" environment, many of the advantages that seem to pertain to the Regular interface would be available to the other interfaces as well. In addition, it is instructive to note the areas where, even *without* the extra degree of freedom, the other interfaces outperformed the Regular interface, particularly regarding the number of pages visited.

It is evident that each interface has good and bad features. Recency fares well in terms of the rate of "getting lost"; users found what they wanted within a reasonable amount of time. However, it was not significantly faster than the others for those who *weren't* lost. It also seems to require users to try more pages and to hover over more pages to reach the desired target. The latter can easily be explained by the enormous menu that results after some time browsing. We later decided that we should have added one more Task, "return to main page"; we observed users being far more confused when asked to do this with Recency than with the other interfaces. This is because the "main page" in the Recency interface is *not* always the least recent page, whereas it is always the last item in the other interfaces.

We had hoped that Histree would have been the most natural to use; after all, it duplicates the already well-known Regular interface but simply adds the ability to recreate one's full navigational path. However, users often did not pay close attention to their navigational paths, and were confused by the re-drawing of the tree as they moved around the menu. They also were inexperienced with clickable submenu headers, causing them to constantly follow submenus rather than concentrate on the headers. This resulted in abnormally high times and rates of "getting lost". However, Histree still outperformed the others in terms of the number of pages viewed (which we feel to be the most useful measurement in terms of understanding the utility of the algorithm) and number of items hovered over.

FamilyTree seemed to take Histree's advantages and add to them, surprisingly. In terms of time and "getting lost", it appreciably outperformed Histree, and did not notably differ in the other variables. There are several possible reasons for this. Firstly, FamilyTree allows every single page to be accessible from the Back menu – no Forward menu is necessary. It also does not have any duplicate entries (internally), meaning that a user can *always* return to a child from the parent.

We believe that the strengths of Recency and FamilyTree can be combined by changing FamilyTree to have its first level be drawn as a recency list rather than a stack. This allows users to take advantage of Recency's linearity and ease of use, while still allowing them to fully explore their navigation.

REFERENCES

Ayers, E. Z. and Stasko, J.T., 1995, Using Graphic History in Browsing the World Wide Web. *Proceedings of the Fourth International World Wide Web Conference.*, pp. 1-8.

Catledge, L. D. and Pitkow, J.E., 1995, Characterizing browsing strategies in the World-Wide Web. Proceedings of the Third International World-Wide Web conference on Technology, tools and applications. Darmstadt, Germany, pp. 1065-1073.

Cockburn, A. and Jones, S., 1996. Which way now? Analyzing and easing inadequacies in WWW navigation. *International Journal of Human-Computer Studies,* Vol. 45, No. 1, pp. 105-129.

Cockburn, A., et al., 1999, WebView: A Graphical Aid for Revisiting Web Pages. *OzCHI'99: Australian Conference on Computer-Human Interaction.*, pp. 7-14.

Cockburn, A. and McKenzie, B., 2001. What do web users do? An empirical analysis of web use. *Int. J. Hum.-Comput. Stud.,* Vol. 54, No. 6, pp. 903-922.

Cockburn, A., et al., 2002. Pushing Back: Evaluating a New Behaviour for the Back and Forward Buttons in Web Browsers. *International Journal of Human-Computer Studies,* Vol. 57, No. 5, pp. 397-414.

Cockburn, A., et al., 2003. Improving Web Page Revisitation: Analysis, Design and Evaluation. *IT&Society (www.itandsociety.org),* Vol. 1, No. 3, pp. 159-183.

Doemel, P., 1995. WebMap: a graphical hypertext navigation tool. *Comput. Netw. ISDN Syst.,* Vol. 28, No. 1-2, pp. 85-97.

Frecon, E. and Smith, G., 1998. WebPath - A Three-Dimensional Web History. *infovis,* Vol. 00, No., pp. 3.

Gandhi, R., et al., 2000, Domain Name Based Visualization of Web Histories in a Zoomable User Interface. Proceedings of the 11th International Workshop on Database and Expert Systems Applications., pp. 591.

Greenberg, S. and Cockburn, A., 1999, Getting Back to Back: Alternate Behaviors for a Web Browser's Back Button. *Proceedings of the 5th Annual Human Factors and the Web Conference.* NIST, Gaithersburg, Maryland, USA.

Greenberg, S., et al., 2000. *Contrasting Stack-Based and Recency-Based Back Buttons on Web Browsers*. Department of Computer Science, University of Calgary, Calgary, Report 2000-666-18.

Hightower, R. R., et al., 1998, Graphical multiscale Web histories: a study of padprints. *Proceedings of the ninth ACM conference on Hypertext and hypermedia.* Pittsburgh, Pennsylvania, United States, pp. 58-65.

Jhaveri, N., 2004. Intermediate and Post-Session Web Page Revisitation Techniques and Tools. In *Computer Sciences*, pp. 53.

JasonSmith, M., and Cockburn, A., 2003, Get a way back: evaluating retrieval from history lists. *Proceedings of the Fourth Australian user interface conference on User interfaces 2003 - Volume 18.* Adelaide, Australia, pp. 33-38.

Kaasten, S. and Greenberg, S., 2001, Integrating back, history and bookmarks in web browsers. *CHI '01 extended abstracts on Human factors in computing systems.* Seattle, Washington, pp. 379-380.

Kaasten, S., et al., 2002, How People Recognize Previously Seen Web Pages from Titles, URLs and Thumbnails. *People and Computers XVI (Proceedings of Human Computer Interaction 2002).*

Milic-Frayling, N., et al., 2003, WebScout: Support for Revisitation of Web Pages Within a Navigation Session. *Proceedings of the IEEE/WIC International Conference on Web Intelligence.*, pp. 689- 693.

Milic-Frayling, N., et al., 2004, SmartBack: Supporting Users in Back Navigation. *Proceedings of the 13th international conference on World Wide Web.* New York, NY, pp. 63-71.

Orner, D. and MacKenzie, I.S., 2006, Histree: A hierarchical back menu. *Proceedings of IADIS International conference WWW/Internet 2006 (to appear).*

Robertson, G., et al., 1998. Data mountain: using spatial memory for document management. In Proceedings of the 11th annual ACM symposium on User interface software and technology, pp. 153-162.

Tauscher, L. and Greenberg, S., 1997. How people revisit web pages: empirical findings and implications for the design of history systems. *International Journal of Human-Computer Studies,* Vol. 47, No. 1, pp. 97-137.

Wen, J., 2003. Post-Valued Recall Web Pages: User Disorientation Hits the Big Time. *IT&Society (www.itandsociety.org),* Vol. 1, No. 3, pp. 184-194.