Cite as:
Tauscher, L. and Greenberg, S. (1996) Design Guidelines for Effective WWW History Mechanisms. In Microsoft Workshop, Designing for the Web: Empirical Studies. Microsoft Corporation, Redmond, WA. October 30.

Design Guidelines for Effective WWW History Mechanisms

Linda Tauscher and Saul Greenberg
Department of Computer Science, University of Calgary
Calgary, Alberta CANADA T2N 1N4
Tel: +1-403-220-6087
E-mail: tauscher *or* saul@cpsc.ucalgary.ca

ABSTRACT

This paper presents design guidelines for history mechanisms within graphical World Wide browsers, and describes the methodology used to formulate them. Our hypothesis is that users revisit World Wide Web (WWW) pages, and that an examination of individual's WWW navigation patterns can provide insight into the design of history systems. Data was collected from 23 subjects who used an instrumented version of Xmosaic 2.6 for six weeks. The data was analyzed in three ways. First, we assessed the extent to which pages are revisited. Second, we examined five possible patterns of page reuse. Third, we applied various *conditioning methods* for history lists to evaluate current and alternative approaches. From these empirical results combined with previous research into history systems, nine design guidelines for graphical WWW browser history mechanisms were derived. These guidelines address the following: access to previously visited pages; cognitive and physical effort of using a history mechanism; strategies for offering pages for selection; and end-user customization of the history data.

Keywords

History mechanisms, WWW, hypertext, navigation, design guidelines.

1. INTRODUCTION

The World-Wide Web (WWW) hypertext system is a large, distributed repository of information. People use graphical browsers to navigate through links and to view pages. Within these browsers, history mechanisms allow people to revisit pages they have viewed previously. If people revisit pages often, such history mechanisms can mitigate three problems people face when navigating the WWW. First, they can help the user navigate through the vast amounts and poor structure of information by providing easy access to information previously visited. Second, they can decrease resource use by supplanting search engines for finding old pages, and by eliminating navigation through intermediary pages en-route to the page of interest. Third, they can reduce a user's cognitive and physical navigation burdens. Pages can be returned to with little effort, and they can show users where they have been.

However, today's design of history mechanisms tend toward ad-hoc approaches that do not appear to take advantage of previous research into history support within user interfaces e.g. Greenberg (1993) and Lee (1992). In particular, their designs are not based upon actual studies of how people revisit Web pages, and their actual use has been examined only superficially.

Our goal is to place the design of history mechanisms within browsers on a more empirical footing. We had three sub-goals.

1. We wanted to understand people's revisitation patterns when navigating the WWW, yet little empirical data is available. The proportion of Web pages that are revisited by a particular user has not been quantified, and no research has examined patterns of page reuse. Section 3 summarizes our quantitative results about revisits to Web pages and five possible patterns of page reuse (Tauscher, 1996a).
2. We wanted to evaluate current approaches in today's history systems, validate successful solutions, and suggest better alternatives. Yet today's history mechanisms are rarely evaluated. From research by Catledge and Pitkow (1995) we know that *Back* is heavily used to return to a page, but the history list is not. Cockburn and Jones (1996) performed a usability study that illuminated user's difficulties with the current stack-based history mechanism. Section 4 summarizes our evaluation of the goodness of predictions offered by this and other schemes.
3. We wanted to provide guidelines to facilitate effective browser history design. Section 5 describes nine design guidelines that are based upon our empirical results, and previous research into user interface history support.

2. DATA COLLECTION

XMosaic 2.6 was modified to record a user's browsing activity. Each activity record included time, URL, page title, final action, method of invoking action, user id, and other items. Volunteer participants then used the browser for six weeks; all were practiced Web users with at least one year of experience. At the end of the study, we analyzed logs from 23 participants. This was followed by hour-long individual interviews, done to gather qualitative data about personal browsing methods and to help us understand why the patterns seen in the analysis arose.

3. RESULTS

Six analyses pertaining to the reuse of Web pages are presented here. First, we report the rate that Web pages are revisited. The remaining analyses concern five different patterns that may suggest effective approaches to presenting revisited pages for future access. For the first pattern, we examine whether users continuously and uniformly visit new Web pages over the duration of their browsing episodes. Second, we look at the distance (in terms of URLs) between repeated visits to the same URL. Third, we assess the frequency of URL visits. Fourth, we determine the extent to which users browse within clusters, or locality sets, of related pages. Last, we identify repeated sequences of URLs as an estimate of path-following behaviour.

3.1 Recurrence of Web page visits

History systems are only useful if users actually repeat their activities. While Web browsers contain a history mechanism, we do not know how often people revisit their pages. In other domains, research has quantified this repetition of user actions e.g. telephone numbers dialed (57%), information retrieved in a technical manual (50%), and Unix command lines entered (75%) (Greenberg, 1993). We analyzed our own data and the Catledge and Pitkow (1995) data to derive a *recurrence rate R:* the probability that any URL visited is a repeat of a previous visit.

An overall recurrence rate of 58% (σ = 9%) shows that users do revisit Web pages, and qualifies Web browsing as a *recurrent system*—one in which users predominately repeat activities they had invoked before (Greenberg, 1993). Yet, users do not fixate on a small set; they incorporate new pages into their repertoire as well. This seems to be a generalizable result, for we found a mean recurrence rate of 61% (σ = 9%) in the Catledge and Pitkow data.

Post-study interviews gave us the opportunity to learn more about users' reasons for visiting new pages, and revisiting old ones. We found that people revisit pages because the information contained by them changes, they wish to explore the page further, the page has a special purpose (e.g. search engine, subject index, home page), they are authoring a page, or the page is on a path to another revisited page. People visit new pages because their information needs change, they wish to explore a particular site, a page is recommended by a colleague, or they notice an interesting page while browsing for another item.

3.2 Growth of URL vocabulary

This pattern of page reuse shows the distribution of old and new page visits over time. We generated vocabulary graphs for each participant, where the URL vocabulary—the number of unique URLs visited so far —is plotted over the total number of URLs visited. For example, Figure 1 shows

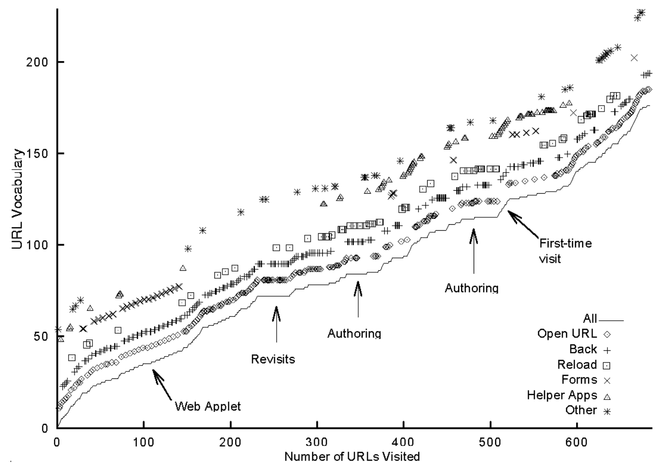

Figure 1. URL Vocabulary for participant 15

the plot for participant 15. The *All* curve represents the overall URL vocabulary size at any moment in time. Major navigation actions are also plotted as separate curves shifted above the vocabulary line by a constant amount: *Open URL*, *Back*, *Reload*, *Forms*, *Helper Applications*, and *Other*. The *Other* category includes all remaining navigation actions. These curves when the most common navigation actions were invoked and, taken together, comprise the *All* curve.

URL vocabulary growth curves for all participants exhibit a linear and positive slope typified by the line in Figure 1. Both data and interview results indicate that users incorporate new URLs into their repertoire at a regular rate, and that revisits are fairly evenly distributed. Yet local variations to the slope are also evident, the nature and extent of which vary among individuals. These local variations (and their concomitant navigation actions) highlight several browsing patterns. We identified seven patterns, and four are illustrated in Figure 1.

1. *First-time visits* to a cluster of pages is evident at the steeply sloped area between URLs 510 to 525.
2. *Revisits to pages*. Plateau areas show revisits to pages. For example, the long plateau in combination with the Back or Open URL actions between URLs 240 to 280 occurred when this participant reviewed online course notes.
3. *Authoring of pages*. These manifest themselves as plateaus, where the subject used Reload extensively to view the modified page e.g., between URLs 480 to 510.
4. *Regular use of web-based applications*. Between URLs 50 to 150 (x-axis), there is a moderately sloped area with a combination of Open URL, Back and Forms activity. This was caused by the subject's revisits to a knowledge elicitation tool packaged as a Web application.

5. *Hub-and-spoke*. People visit a central page (hub) and navigate the many links to a new page (spoke) and back again. This is akin to a breadth-first search.
6. *Guided tour*. Some page sets include structured links (e.g., next page), and people can choose to follow these.
7. *Depth-first search*. People follow links deeply before returning to a central page, if at all.

While Web pages are recalled, new pages are incorporated at a regular rate. Thus, removing pages that are unlikely to be revisited is crucial to reducing information overload in a history system. The *pruning* method should also ensure that the remaining items have a high probability of being on the list when the user wants them. This warrants an investigation into the distance between recurrences, which is the next pattern that we examine.

3.3 Web page visit frequency as a function of distance

For any URL visited, the probability that it has already been seen by the user is quite high (58%). But how do particular URLs contribute to this probability? Do all URLs visited have a uniform probability of recurring, or do the most recently visited URLs skew the distribution? If a graphical history mechanism displayed the previous *p* entries as a list, what is the probability that this includes the next entry (Greenberg, 1993)?

The recurrence distribution as a measure of *distance* was calculated for each participant. Distance is determined by counting the number of items between the current URL being visited from its first match on the history list. For example, a distance of one occurs when the user reloads the current page, or successfully interrupts the transmission of a page. A distance of two occurs when the current page is a revisit to the one seen two pages back. Figure 2 plots this data up to a distance of 50, averaged across all participants. The horizontal axis shows the position of the repeated URL on the history list relative to the current one. The vertical axis represents the rate of URL recurrence at a particular distance, denoted as R_d. According to Figure 2, there is a R_{d1} = 10% probability that the current URL is a repeat of the previous URL (distance = 1), R_{d2} = 19% for a distance of 2, R_{d3} = 2%, and so on.. The spikes at distances of 2, 4, 6, and 8 arise from users' navigating back to previous pages by the *Back* navigation action.

However, the most striking feature of the data is the extreme recency of the distribution. The previous 6 or so URLs contribute the majority of pages visited next, although the probability values of R_d continually decrease after the second item. Beyond a distance of 8, the low values (< 1% each) and the low rate of decrease make items equivalent for practical purposes.

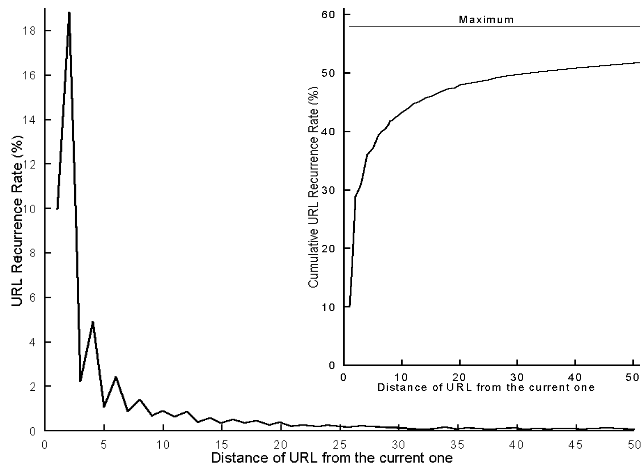

Figure 2. URL recurrence rate as a function of distance (all participants); inset plots *R* as running sum

This is illustrated further in the inset of Figure 2, which reports the same data for all subjects as a running sum of the probability, denoted as R_D. The most recently visited URLs are responsible for most of the cumulative probabilities. For example, there is a R_D = 39% chance that the next URL visited will match a member of a set containing the 6 previous submissions.

3.4 Frequency of URL accesses

Frequency is a popular method for ranking items of interest. We examined this pattern in two ways. First, we generated a frequency graph for each subject. Second, we developed a taxonomy of conceptual page types for frequently visited pages from our post-study interviews. All subjects produced a similar frequency distribution where only a small number of URLs are highly visited, and a very large number of URLs have very low usage frequencies. Over all subjects, 60% of pages were only visited once, 19% were visited twice, and 8% were visited three times. The few pages that were frequently accessed tend to fall into certain categories that also explains their popularity: personal pages, start-up documents, indices, search engines, individual and organization home pages, Web applications, navigation pages, and authored pages.

3.5 Locality

While recency characterizes recurrences in terms of distance, locality characterizes recurrences in terms of periods of time where references are made solely to a small and related group of items, called a locality set (Lee, 1992). We applied the locality detection algorithm (Madison and Batson, 1976) to the WWW data to determine whether users generate locality sets, that is, whether they browse within clusters of pages.

While locality sets were found (Tauscher, 1996a), they do not appear to offer much value in terms of predicting the user's next activity within Web browsing. There are several reasons for this claim. First, most locality sets were very small consisting of only one or two unique URLs. Second, these sets lasted for only a short time (usually 2.5 to 4.5 URLs). Third, few locality sets were repeated; sets that were repeated tended to be of size one or two. Fourth, only 15% of URLs visited were part of a locality set.

3.6 Longest Repeated Sequences

The concept of *paths*, an ordered traversal of hypertext links, has been associated with hypertext ever since Vannevar Bush envisioned hypertext in 1945. If paths exist, it may be useful to capture and offer them via history, thus simplifying people's efforts to retrace a path. Also, if users follow paths solely as a route to a destination, shortcuts could allow a user to go directly there.

We applied the Pattern Detection Module algorithm to the WWW browsing data in an attempt to identify longest repeated sequences (LRSs) of page visitations. As with locality, we discovered that LRSs are not particularly useful for predicting Web browsing for several reasons. We found that though LRSs do exist, they tend to be short (Tauscher, 1996a). The few longer LRSs usually reference only one or two pages. In terms of repetition, the average frequency for LRSs of all lengths hovered around two which is the minimum requirement to be considered a LRS. Also, there is a strong recency effect: repeats of LRSs occur within a short distance of each other.

4. CONDITIONING THE DISTRIBUTION

The recurrence distributions were derived by considering all page visits for a user as one long input stream with no barriers placed between sessions. We have seen in Section 3.3 that although a small set of recently visited URLs accounts for a high proportion of revisits, others lie outside. Consider a set of the 10 previous URLs on the history list: there is a 42% chance that the next URL has not appeared before, a 43% chance that it has occurred within the set, and a 15% chance that it last appeared further back (Tauscher, 1996a). This section explores the possibility that the distribution can be conditioned, first to increase the recurrence probabilities over a set of a given size, and second to evaluate methods that are currently in use. Eight conditioning methods are discussed within four major categories: recency, frequency, stack, and hierarchically structured. A later results method summarizes how effective each method is.

4.1 Recency ordered history lists

Three types of recency ordered history lists were evaluated. The first is *sequential ordering,* the time-ordered list of all URLs visited by the user, including revisits to the same URL. These duplicates occupy valuable space on a history list of a limited length. Hence, two strategies for pruning redundant URLs were applied: saving the URL in its *original position* on the history list, and saving the URL in its *latest position*.

4.2 Frequency ordered history lists

Frequency ordering, where the most revisited page appears at the top of list and the least visited page appears at the bottom, is perhaps the most obvious way of ranking URLs. An issue associated with frequency ordering is how to break ties, that is, how to order URLs that have the same frequency. Greenberg (1993) evaluated two schemes for *secondary sorting* within frequency ordered lists: recency and reverse-recency. Recency was found to perform better so that is the method of secondary sorting that we have applied.

4.3 Stack-based approaches

Current Web browsers maintain a history list that operates as a stack to present the linear path from the first URL visited to the current URL. In many browsers, the most recent page appears at the top of the list while the least recently accessed page appears at the bottom. This is a simplistic and somewhat inaccurate description for there are some nuances in how the current history mechanism operates. Understanding the subtleties of browser history lists requires differentiating the three classes of user technique for page display as described by Cockburn and Jones (1996): *loading*, *recalling*, and *revisiting*. Loading a page causes it to be added to the top of the stack, possibly resulting in all pages above the current position to be lost. Recalling a page merely changes the *pointer* to the currently displayed page in the history list. Revisiting a page occurs when the user explicitly reloads the current page, and has no effect upon the history list.

A variation on the stack method is the persistent stack—it makes the stack at the end of the previous browsing session for each browser window available for the next session.

4.4 Hierarchically structured history lists

Two methods that employ hierarchical structuring were examined: recency ordered hyperlink sublists, and context-sensitive Web subspace history lists. The first method is similar to the recency ordered history list in which duplicates are pruned and saved in their latest accessed position. The difference is that for each URL on the normal list, a secondary list of the hyperlinks visited from that URL can be raised. The user first scans down i entries in the normal list for an exact match that terminates the search, or for an entry that contains the desired hyperlink. In the latter case, the sublist of hyperlinks is displayed (perhaps as a cascading menu) and the search continues until an exact match is found j entries later. The distance of a matching recurrence is simply $i + j$.

The second method, context-sensitive *Web subspace* history lists, is based upon a graphical history display designed by Cockburn and Jones (1996). The display creates a new Web subspace each time the user *directly accesses* a page. This page is added to the *Webs* menu, and any pages accessed within this subspace are added to a cascading or secondary

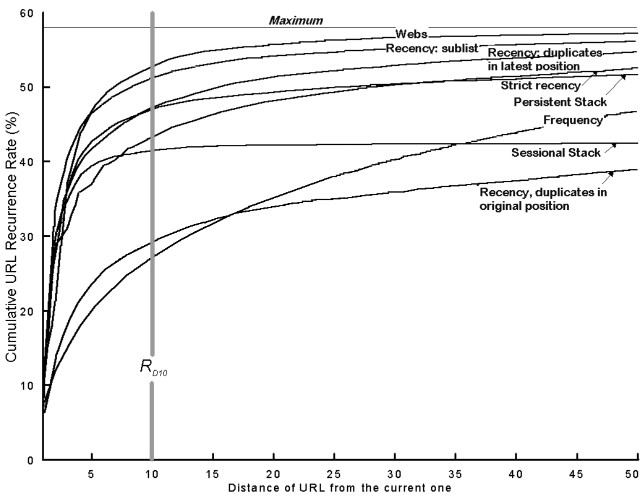

Figure 3. Cumulative probabilities of recurrences over distances up to 50

menu that is linked to the menu entry for the first page in the subspace. For our analysis, we considered the following actions as a *direct access* to a URL: typing a URL, selecting a Hotlist item, cloning or opening a new window, and accessing a URL via client-dependent hard-wired buttons or menus. Within the main and secondary menu, we sort the URLs based on recency, and remove duplicates, saving according to the latest position.

4.5 Results

We evaluated all methods described above by implementing them as algorithms, and using our subjects' traces to simulate their performance in practice. Though our analysis does not compare a user's cognitive and physical effort involved in choosing items from each history method, we do discuss (but not measure) the simplicity of the method's conceptual model.

1. The benchmark method, a strict sequential list of URLs ordered by recency performed reasonably well with a small set of URLs e.g. R_{D10} = 43%. A benefit of this method is that its conceptual model is simple and familiar—a person knows what they have just done and can thus predict if an item will be on the history list. Also, recency methods do not suffer from the initial startup instability that other methods do when there are only a few URLs available to present to the user.

2. Pruning duplicates is a simple way of improving the performance of a recency-ordered list when duplicates are saved in their latest versus original position (R_{D10} = 47% vs. 43%). The former approach performs better because just visited URLs will stay at the top of the list, and because local context is maintained (Greenberg, 1993). We expect that this type of list still presents a clear conceptual model, though it does not the exact sequence of URLs visited by the user. The striking differences between the three recency orders are illustrated by the plots in Figure 3.

3. Frequency ordering is the worst predictor of the 8 evaluated for short lists with R_{D10} = 27% vs. 43%. While it does improve as distance is increased, it does not catch up to strict recency. Frequency has other problems. Users may find it more difficult to predict which pages would appear on a frequency-ordered list beyond the two or three that they visit the most. As well, frequency ordering suffers instability when few items are on the history list, and excessive inertia when the list is long. Still, frequency could be applied to a few key URLs, possibly as an auxillary method in conjunction with another history mechanism that gives better overall performance.

4. The sessional stack method found in most Web browsers is slightly better than strict recency at very short distances (e.g. R_{D5} = 40% vs. benchmark of 37%) and worse at R_{D10} (42% vs. 43%). As seen in Figure 3, it is much worse as the list gets long. This is because URLs are not retained between sessions, and some recent URLs are removed when the user loads a page while at a point other than the top of the stack. Also, there are problems with this method's conceptual model, as Cockburn and Jones (1996) discovered in their usability study.

5. The persistent stack is an improvement over the stack method in terms of its recurrence probabilities over distance because some URLs are retained between sessions. However, it still suffers from the conceptual model difficulties that the stack method does. Also, the persistent stack can potentially contain many duplicate entries over time, and it may not contain all URLs visited.

6. Recency ordered hyperlink sublists have the highest recurrence probability over all methods for very short distances (2-4), and are second best at modest distances (R_{D10} = 51% vs. 43%). This method makes more URLs accessible (up to 55 with a main list of 10 items), and allows the user to select a hyperlink on a page without visiting the page or locating it on the current page. However, this result is optimistic as greater cognitive and physical effort is involved in selecting items from the hyperlink sublists e.g. to make an accurate selection from a hyperlink sublist, the user must recall which main list item contains the desired URL.

7. The best method we evaluated—context-sensitive web subspaces—showed that 53% of all URL selections can be successfully predicted with a set of 10 items. Given that R = 58% on average, which is the best a perfect reuse facility could achieve, this method is potentially about 91% effective. However, context-sensitive web subspaces suffer from the same problems that recency

ordered hyperlink sublists do. They require greater physical effort to select a sublist item, and greater cognitive effort to recall which sublist might contain the URL. In addition, the context-sensitive web subspaces method's conceptual model may be more difficult to comprehend. Users need to understand the notion of a 'direct access URL' to grasp the way the history list is organized.

5. DESIGN GUIDELINES

This section proposes nine guidelines for the design of history mechanisms in WWW browsers. They are derived from those formulated by Greenberg (1993) for the design of reuse facilities, and from our empirical results: recurrence rate, five patterns of revisitation, and conditioning methods analysis.

1. *Maintain records of URLs visited, and allow users to recall previous URLs from those records*. Our study shows that though users incorporate new URLs into their repertoire at a regular rate, 58% of Web pages are revisited. Web navigation is thus classified as a recurrent system. Hence, a history mechanism has value, and as a first requirement, it must record the URLs that users visit. To obtain the maximum benefit from this data, users must be able to access the URLs during their current as well as later sessions.

2. *It should be cheaper, in terms of physical and cognitive activity, for users to recall URLs from a history mechanism than to navigate to them via other methods*. The prime motivation for providing a history system is to reduce the physical and/or cognitive effort of returning to a particular Web page. We found that users select URLs from Mosaic's Window History dialog less than 1% of the time, likely due to various physical and cognitive overheads involved. Several factors in history mechanism design affect its ability to reduce the overhead in returning to a Web page.

    - The history system should attempt to predict the user's next URL selection. If it does so effectively, the user is likely to access the history system to retrieve the URL versus navigating to it via other methods.
    - The best predictions should be clearly distinguishable so that they are the first ones that the user sees. For example, the most likely history items could be placed at the top of the list or they could be presented as top level buttons (*Back* works reasonably well because of this).
    - A minimum number of physical actions should be required to access and retrieve an item from the history system.
    - The history mechanism should provide some clues as to the structure of the Web space and the pages previously visited to help the user regain context and orient themselves.
    - Accessing the history mechanism should be minimally disruptive to the user's current task.

3. *A selectable history list of the previous 10 URLs visited provides a reasonable set of candidates for reuse*. Greenberg (1993a) concludes that a lengthy history list is unlikely to be worthwhile considering the high cost of real estate on even large screens, and the user's cognitive overhead of scanning the possibilities. For example, our results show that a menu of the previous 10 URLs visited covers, on average, 43% of all inputs. Doubling this to 20 items only increases the probability to 48%. However, the list could be even shorter than 10 URLs since the items that contribute most to the probability of a recurrence are at a distance of one, two and four (10%, 19%, and 5% respectively). Another benefit of presenting the most recent URLs is that the user will likely be able to predict if the URL they seek will appear on the list.

4. *Other strategies for presenting the history list, particularly pruning duplicates and hierarchical structuring, increase the probability of it containing the next URL*. A significant number of URLs are not covered by the last 10 items (26% of the recurring total) though doubling or tripling the size of the list does not increase its coverage much. But it is these URLs that could help the user most since they occurred long ago and are thus more difficult to recall and/or locate. This is why it is important to explore alternative methods for conditioning the history list. Pruning duplicates from a recency ordered list is a simple improvement. Also, hierarchical structuring is becoming a popular method of presenting history (Tauscher, 1996a).

5. *History based on recency is not effective for all possible recalls because it lists only a few previous events. Alternative strategies must be supported.* Recency was a strong reuse pattern but we found that other patterns exist. For example, a few key pages are accessed with a high frequency. One of these, the user's home page, is easily accessible by the option of it being the start-up document, and the Home button on the browser toolbar. Other frequently accessed pages could be made available on a toolbar for easy access. A drawback of frequency ordering is that it has a certain degree of non-intuitiveness. That is, during post-study interviews, subjects were sometimes surprised to see certain URLs on their 15 most frequent URLs list. Greenberg (1993a) suggests that combining a recency-based short-term memory with a frequency-based long-term memory could generate better predictions. For example, the browser could show the most recent

URLs, as well as the top 3 most frequently visited pages.. Two other alternative strategies are worth mentioning. First, identifying and presenting paths to the user may be useful though additional research is required to improve path detection within the WWW domain. Second, for infrequently accessed URLs that have not been visited recently, the ability to search one's history could be beneficial.

6. *URLs already recalled through history should be easily reselectable*. If a user has selected an item from their history, the item is probably of more importance to them. Thus, it should be easier for them to retrieve that item in the future. This goal can be facilitated implicitly and/or explicitly. For example, certain conditioning methods favour the reselected item by propagating it to the top of the list based on recency of access, or increasing its access count for frequency ordering. Explicit methods for supporting this guideline might highlight the item on a list or in a graphical overview to show its selection during the current browsing session.
7. *History items should have a meaningful representation*. Semantic information about a history item is necessary to enable the user to easily locate the item whether it appears on a list or in a graphical display of some sort. Web pages are typically referred to by their URL or their HTML title tag. There are several problems with using the title tag: it may be absent, it may not be the same as the page title (which is usually an *H1* tag), and it may be too long to display easily. URLs, on the other hand, may be long and non-intuitive, and thus difficult to recall, type and/or parse.
8. *Support grouping of URLs into high-level Web tasks, and switching between tasks*. Hypertext encourages connections between information that is related in some way. Thus, a sequence of pages that a user browses may have a particular context that would be convenient to present to the user at a higher-level. We explored this concept in several ways that include identifying locality sets and longest repeated sequences, and evaluating the predictiveness of context-sensitive web subspaces. The latter proved to be the best conditioning method of the eight examined. Locality sets and paths may hold promise if their algorithms were to consider more domain knowledge.
9. *Allow end-user customization of history data*. We believe that the automatic capture of history data is essential to reduce the physical and cognitive overhead of recording URLs for reuse. However, users may also want to customize the various attributes of a history mechanism or save portions of their history. If users are to take advantage of this feature, it is essential that the physical and cognitive overhead of managing history data be kept to a minimum.

CONCLUSIONS

This paper provides empirical data that justifies the need for suitable history mechanisms in graphical Web browsers. Furthermore, an analysis of different designs proves that the predictiveness of the current stack-based model can be improved upon. Using the methodology and design guidelines herein, designers can refine current history mechanisms and investigate new approaches.

There are still many unanswered questions. We have not evaluated the physical and cognitive effort for reviewing a particular conditioned set of history list predictions. Also, we have not assessed the impact of different browser and HTML artifacts upon reuse such as *frames* although we suspect that the numbers reported here would not change dramatically. A third area for future research is validating the design guidelines that we have proposed.

REFERENCES

Bush, V. (1945). As we may think. *Atlantic Monthly*, 176(1), pp. 101-108, June.

Catledge, L. and Pitkow, J. (1995). Characterizing Browsing Strategies in the World-Wide Web. In *Proceedings of the Third International World Wide Web Conference*, http://www.igd.fhg.de/www/www95/ papers/80/userpatterns/UserPatterns.Paper4.formatted. html, Darmstadt, Germany.

Cockburn, A. and Jones, S. (1996). Which Way Now? Analysing and Easing Inadequacies in WWW Navigation. *Int. J. Man-Machine Studies*, 44, submitted for publication.

Greenberg, S. (1993). *The Computer User as Toolsmith: The Use, Reuse, and Organization of Computer-based Tools*. Cambridge series on human-computer interaction. Cambridge University Press.

Lee, A. (1992). *Investigations into History Tools for User Support*. Ph.D. Thesis, Department of Computer Science, University of Toronto, Ontario, Canada.

Madison, A. and Batson, A. (1976). Characteristics of program localities. *Communications of the ACM*, 19(5), 285-294.

Tauscher, L. (1996a). Evaluating history mechanisms: An empirical study of reuse patterns in World Wide Web navigation. MSc Thesis, Department of Computer Science, University of Calgary, Alberta, Canada. May. http://www.cpsc.ucalgary.ca/grouplab/papers/

Tauscher, L. and Greenberg, S. (1996) Revisitation Patterns in World Wide Web Navigation. Research report 96/587/07, Department of Computer Science, University of Calgary, Canada. March. http://www.cpsc.ucalgary.ca:80/projects/grouplab/ papers/96WebReuse/TechReport96.html