

PageLinker: Integrating Contextual Bookmarks within a Browser

Aurélien Tabard^{1,2}
tabard@lri.fr

Wendy Mackay^{1,2}
mackay@lri.fr

Nicolas Roussel^{2,1}
roussel@lri.fr

Catherine Letondal³
letondal@pasteur.fr

¹INRIA
F-91405 Orsay, France

²LRI - Univ. Paris-Sud & CNRS
F-91405, Orsay, France

³Institut Pasteur
F-75015, Paris, France

ABSTRACT

PageLinker is a browser extension that contextualises navigation by linking web pages together and allows navigation through a network of related web pages without prior planning. The design is based on extensive interviews with biologists, which highlighted their difficulties finding previously visited web pages. They found current browser tools inadequate, resulting in poorly organised bookmarks and rarely used history lists. In a four-week controlled field experiment, PageLinker significantly reduced time, page loads and mouse clicks. By presenting links in context, PageLinker facilitates web page revisitation, is less prone to bookmark overload and is highly robust to change.

Author Keywords

Biologists, Bookmarks, Browsers, Contextual bookmarks, PageLinker, Participatory design, Web Navigation, WWW

ACM Classification Keywords

D.2.2 [Design Tools & Techniques]: User Interfaces. H.5.2 [User Interfaces]: Evaluation/methodology, User-centered design. H.5.4 [Hypertext/Hypermedia]: Navigation, User Issues

INTRODUCTION

The World Wide Web has expanded dramatically in the past decade, with huge increases in the number of users, web pages and complexity of content. Unfortunately, at the level of user interaction, web browsers have not kept pace. Early user aids for finding previously visited pages, e.g., bookmarks and history, have evolved little since their introduction in the early 1990's [3]. Even though revisitation accounts for half or more of visited pages [6,9,25] studies show that revisitation tools are rarely used [6,25,27].

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

CHI 2007, April 28–May 3, 2007, San Jose, California, USA.

Copyright 2007 ACM 978-1-59593-593-9/07/0004...\$5.00.

We have been studying a particularly web-intensive group of users, research biologists, who have reorganised their work around the internet. They treat the Web as an enormous, constantly searched database and also as an analysis tool. They repeat collections of tasks, revisiting the same sets of pages over and over again, browsing sequentially and in parallel as they analyse data sets and pursue hypotheses. Unfortunately, their improvised, fluctuating workflow is often poorly supported by the websites they use. The data pages they seek may require long navigation paths through huge hierarchical directories, and are unlikely to contain direct links to the analysis programs they will apply to this data. For them, as others, bookmarks and history pages are insufficient.

How can we facilitate page revisitation tasks? Automation tools that allow users to build and play common scenarios offer one solution. However they usually require too much advance planning: Biologists must rethink their workflow at each navigation step and each decision depends upon multiple situated factors [23], including time available, knowledge of server loads or difficult-to-articulate factors such as one's intuitions about whether certain results are 'normal'. Visualisation tools that graphically illustrate previous navigation steps are another possibility. However these require a great deal of screen real estate and focus attention away from the primary navigation task.

Based on these problems, observed with current browsers and other navigation tools (visualization, automation), we developed PageLinker, a browser extension that allows users to *contextualise* their navigation by associating web pages together, i.e. to create and present links only on specific pages or set of pages.

We describe our preliminary study of biologists at several research institutions, with insights gained from interviews, observations, brainstorming sessions and workshops. We then discuss implications for the design of contextual bookmarks, including a review of the relevant literature. We next present the evolution of PageLinker and describe a controlled field experiment to evaluate it. We conclude with an analysis of the results and discuss implications for future research.

PRELIMINARY STUDY

We have been working closely with research biologists for the past eight years [15] in a variety of participatory design projects. Although not necessarily computer programmers, these biologists are highly experienced internet users who have modified their work practices to take advantage of the wealth of biological data and analysis programs available on the web. This study reported here focuses on the problems they face when navigating the web.

We selected 20 biologists who had recently used on-line biological data and analysis programs as an integral part of their research. We conducted videotaped interviews in their labs and asked them to replay, with their own data, their latest analysis using a web application. We also asked them to search for specified information in their research institution's online documentation. We used a talk-aloud protocol, asking them to explain their choices and what bothered them, as they performed these tasks. We also organised a video-brainstorming [19] workshop that focused on organising analysis workflows, either by using and possibly editing history data or by assembling analysis resources on the Web.

Illustrating the navigation problem

The following scenario illustrates a typical navigation session for a biologist studying a protein. Ann needs to explore alternative hypotheses before conducting a time-consuming lab experiment. She begins by collecting data: From the Biology department's homepage, she follows links to the protein database page. Unfortunately, it doesn't offer links to relevant analysis tools and she must browse a huge, hard-to-navigate hierarchical directory with hundreds of links spread over many pages. She eventually finds the relevant page and checks the research literature to see if similar forms of the protein appear. She then looks for the protein sequence in two different databases to find out if different DNA sequences are associated with the protein. She encounters incompatible data formats, forcing her to transform the data before using her chosen analysis protocol.

The lack of relevant links in the data pages makes it difficult for Ann to move from one step to the next. Even when she does find appropriate online resources, she has trouble keeping track of them. Several weeks later, when she decides to analyse a new set of data, she has to recreate her initial search process in order to find the same pages again. Like others in our study, Ann rarely uses bookmarks or history pages, and instead relies on Post-it™ notes, e-mail and search engines to find previously visited sites.

Observations

Several recurrent themes emerged from our interviews and the workshop on online data management:

Habits: Most biologists:

- have bookmarks but often prefer to use search engines, email and physical Post-it™ notes;

- reach previously bookmarked sites via search engines, because their bookmarks are difficult to browse;
- save temporary results or alternative data formats; and
- rarely customize web forms, even when possible.

Software strategies: Biologists are conservative software users. They prefer a stable and predictable set of tools [18] and tend to use techniques they already know rather than learning a new, potentially better one. Most stay with a single Web server if it provides all the tools they need, even though better tools might exist on other servers. Most biologists are usually skeptical of pipelining and automation tools that support biological protocols, since the learning curve is often steep and the benefits are usually limited.

Interaction: Biologists' purposes and procedures change rapidly. Unlike programming, constructing a biological online protocol is not fully algorithmic and requires human judgment along the way. Biologists check the accuracy or significance of results and decide whether and how to carry out an analysis using complex criteria that would be difficult to automate. A biologist might decide to use different processes, proceed with full data or extract subsets depending upon on the characteristics of the data and her current research goal.

Equivalent objects: Data formats are often incompatible: the output of one tool may not be interpretable as input by another tool. Biologists are thus forced to edit intermediate results and end up managing collections of "equivalent" data objects, including:

- same data in different formats needed by different tools
- different versions of the same data, e.g., two versions of an annotated genome.

Data flows: Biologists create diverse data flows, piping the output of one program into another as well as reformatting, transforming, filtering and extracting data [22]. They use copy-paste to chain these steps, which is not supported by automated tools. Like Tauscher & Greenberg [25], we found that they preferred to replay a path rather than using history to access a specific page.

Related work

Our observations match findings in the research literature with respect to re-visiting web pages and recording and connecting resources over the internet. Tauscher & Greenberg define the Web as a 'recurrent system' [24] and report that 58% of pages are revisited. Weinreich et al. [27] reported 46% and Cockburn & McKenzie [9] reported 81% in their respective studies.

Unfortunately, the history and bookmarks mechanisms provided by browsers are not sufficient to support web page re-visitation [6, 25, 27]. Web browsers provide both short-term (*back* and *forward* buttons) and long-term history mechanisms (global history lists). Although *back* is used relatively frequently (14% of navigation actions), global

history lists are rarely used [6, 7, 24], only about 0.2% of all page requests [27]. Stored history information is usually very limited, capturing only the last time (and possibly the first) a URL was visited. This makes it difficult for users to find a page accessed from a well-known site at a specific date. For example, if a biologist follows a path through a series of websites to fulfill a protocol, and one of those sites is visited later, the earlier path will no longer appear in the history file.

Studies of navigation paths show that bookmarks are not a panacea for solving the problem of page access [1]. The changing nature of the web and users' changing interests [20, 25] often cause classification and relevance problems. Page titles are often obscure or too long to be displayed in a menu [20]. Bookmark lists tend to grow over time as users add new pages without removing unused ones [9], providing "neither a reminding function nor a context of relevance"[13]. If users do not constantly edit and prune their lists, they end up with inappropriate and uninteresting URLs, little better than no bookmarks at all [24].

Graphs of navigation history provide an alternative to history lists [12], situating current activity within previously used paths. However, graphs require additional screen space and force users to shift between their primary browsing tasks and a secondary location task. An interesting alternative is WebView [8], a browser enhancement that integrates several revisitation capabilities into a single display, resulting in a compact revisitation tool. While WebView is promising, it focuses mainly on providing a better interaction with the global history.

Another trend in revisitation tools is to automate navigation. However, the instability of the Web introduces problems: changes in page content, URLs, and data formats can "break" formerly correct automation sequences. Other common problems with automating complex workflows are the lack of transparency, as users search for the cause of unexpected results, and the lack of interaction, when they need to explore possible changes to a sequence. For example, one biologist commented that he "needs to redo the protocol step by step because there is no convenient way to access the problem source directly". The process of navigating through various websites acquaints biologists with changes on the server, new programs, and new layouts that might provide easier access to some pages, helping them to gather knowledge about their virtual environment. Teevan et al. [26] argue, in another context, that directed situated navigation reduces the quantity of information that users need to specify and provides the context they need to help them understand the results they obtain.

Initial design choices

Based on our interviews and insights gained from earlier studies [15, 16], we decided to focus on supporting the biologists' process of analysing web-based data. We wanted to create a tool that fit with their existing work patterns, so they could use familiar work practices and their own data

and not be forced to add additional tasks. We based the design on our observation that biologists use *copy* to extract data from one web page and *paste* to enter it into an analysis form¹, thus identifying which pages make sense to link together.

We selected the Firefox web browser because it is available on Mac OS X, Linux, and Windows and was already used by half the biologists in the study. Installing a Firefox extension is easy: users need only click on the link of the extension they want to install. Firefox can also track *copy* and *paste* events, making it possible to automatically generate the links we observed above. PageLinker takes advantage of this functionality and allows users to contextualise their navigation, automatically linking web pages as the biologist cuts and pastes between them. Later versions of PageLinker also allow users to create these contextual bookmarks manually and offer feedback by showing the most recently created link in the menu.

ITERATIVE DESIGN OF PAGELINKER

Phase 1: Initial implementation

The first version of PageLinker focused on creating links invisibly, based on the user's cut, copy and paste actions. PageLinker overrides copy, cut and paste events: When a *copy* or *cut* event is detected, it records the page (title, URL, and date) and, as soon as a *paste* event is detected, creates a link between the two pages. The *copy* (or *cut*) page thus points to the page where the *paste* occurred. Our interviews and workshops indicated that biologists rarely use output data from one page when they need to fill out a new form. Instead, they usually edit the data, either to address incompatible data formats or to refine their request. We link the page of the most recent copy event to the current paste page, without considering the contents of the clipboard. We can thus accommodate the "equivalent objects" mentioned earlier, where the physical data formats are different but, from the biologist's perspective, the content is the same.

PageLinker uses XUL, JavaScript and RDF². The new definitions of copy, cut and paste items from the menus are implemented with XUL, an XML-based language used to define interfaces. JavaScript handles user interface actions and manages data. We override the clipboard *shortcuts* events by grabbing Ctrl-C/X/V on Windows and Linux or Cmd-C/X/V on Mac OS. We use RDF to implement file recording of contextual bookmarks. A collection of RDF statements represents a labeled, directed graph. Figure 1 shows the graph illustrating a link between two pages. Each page is a node pointing to the pages it is related to. Since RDF allows only simple oriented graphs, our structure is redundant for bi-directional links.

¹ We use the term *form* to refer to pages that require the user to enter data. Some of these forms also generate data.

² See: <http://developer.mozilla.org/>

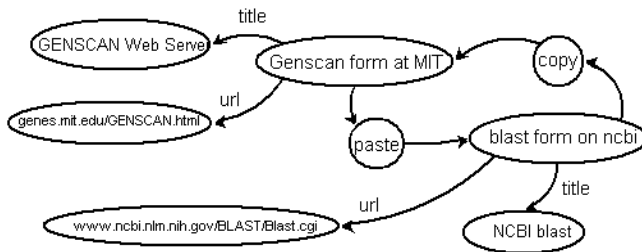


Figure 1: Graph outline of a link between the GENSCAN results (copy) and the BLAST form (paste).

Each page points to its descriptors, e.g., title and URL, as well as a *copy* node and a *paste* node. The copy node points to the list of pages where data was copied from the current page and the paste node points to the list of pages from where data was pasted into the current page. The RDF is queried through a template-based request language supported by XUL in order to map the contextual bookmarks display and the RDF file. When the RDF is modified, its corresponding UI component is automatically updated.

How do we decide which part of the URL to use? If we use the entire URL, the result is too restrictive: we get a large number of pages with only minor variations among them. If we use the root URL, i.e. the main site at the top of a hierarchy of web pages, we only get the main site and lose all of the interim searching the user has done. PageLinker uses the full URL, minus the query string. The resulting contextual bookmarks are specific to a particular web form, rather than a particular result or the whole server.

Iterative design based on user feedback

PageLinker was created using a participatory design process together with biologists at the Institut Pasteur. We tested the first version, PageLinker 0.1, with six biologists who installed it and provided constructive feedback via interviews and direct observation. We chose the simplest design possible: links were based on invisibly-captured copy-paste events and users interacted with PageLinker via the *Shortcuts* menu (Figure 2).

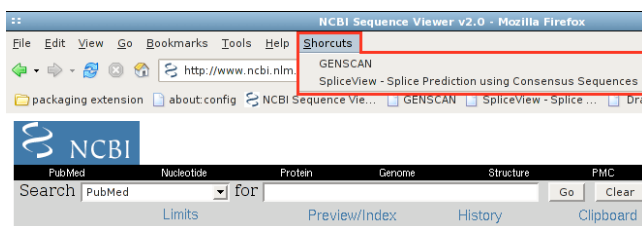


Figure 2: Shortcuts contextual menu (PageLinker 0.1)

Over time, users found that obsolete items had ended up on the Shortcuts menu and asked us to remove them. At this point, some users discovered how to use PageLinker to manually add links between pages, an example of co-adaptive behavior [17]. They used the Ctrl-C and Ctrl-V

shortcuts on pages without entry forms, for example, between one page with press reviews and another with the referenced newspaper articles or between an application form and the relevant documentation page.

Although using control-keys was fine for some users, others requested a more convenient interface for manually linking of contextual bookmarks. Several people liked the concept but found it annoying to copy-paste when it was not required for the task at hand. They commented that they would decide to link back to a previous page only *after* they had successfully identified an interesting subsequent page. Using the copy-paste technique required returning to the previous page and generated meaningless extra actions.

Based on this feedback, we conducted a participatory design workshop to explore simpler ways to create links between pages. We worked together with the biologists to create video prototypes [19] that envision scenarios for linking to a desired destination from a previous page. We created prototypes of three linking strategies: via open pages or tabs, via the last visited page and via the global history.

PageLinker 0.2 implemented all three methods. We added a *link to* menu to the toolbar (Figure 3) that presents a list of all the browser's open web pages (both on tabs and in other windows) and the seven most recently visited websites from the global history. Links are sorted by time, similar to Firefox's *Go* menu. Selecting any of these creates a link from that page to the current page.

PageLinker 0.2 also created a reverse link, from the current page to the one just selected. We reworked the *Shortcut* contextual bookmarks menu to separate links by direction. One list presents links *to* the current page (either via copy-paste or direct selection). The other list presents links *from* the page. Links on both menus were ordered by recency. Based on user requests, we also added the ability to delete a contextual bookmark by right clicking on the corresponding menu item. After one week of use, users said the *link to* menu was too complex and redundant. Bidirectional links presented in two different menus were also too heavy-weight and users did not notice they that could delete them.

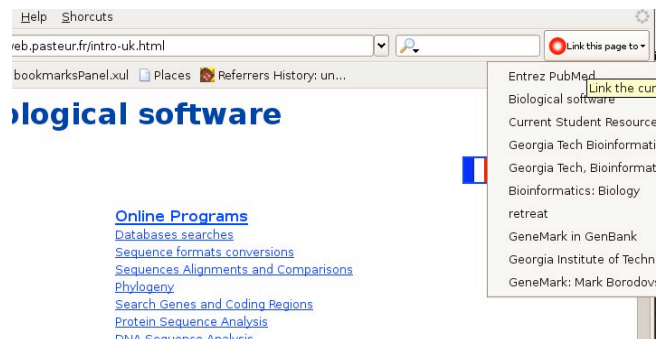


Figure 3: Linking menu prototype (PageLinker 0.2)

PageLinker 0.3 simplified the linking menu to include just the last visited pages. We also classified bidirectional *Shortcuts* by order of recency. Finally, we integrated contextual bookmarks and linking via the bookmarks sidebar (Figure 4). Most users quickly began using the bookmark sidebar instead of the menu. They found it useful to have their contextual bookmarks visible immediately upon changing pages, without needing to click on the menu list, since contextual bookmarks change from one page to the next.

We used PageLinker 0.3 for the field experiment (described in the next section). After the experiment, we released PageLinker 1.0 which included a minor modification: To avoid confusion between the contextual bookmarks list and the linking list, we converted the *link to* list into a menu. Table 1 summarises the four versions of PageLinker, including the types of links, how contextual bookmarks are created and how to access PageLinker.

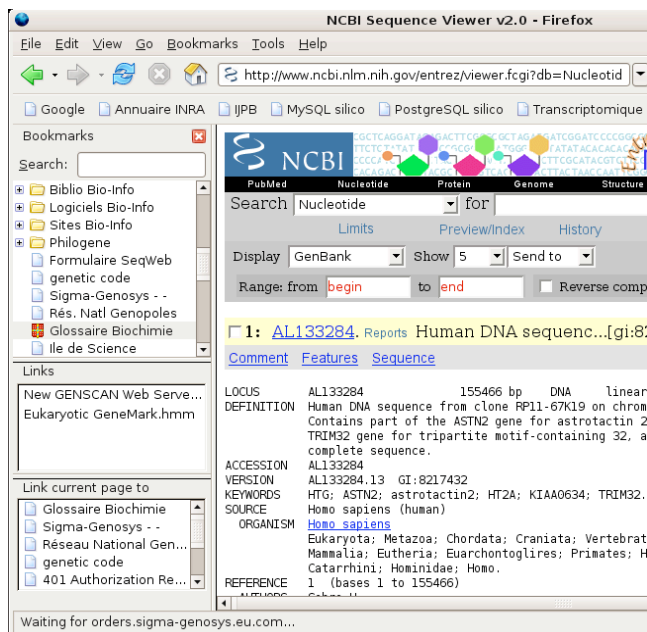


Figure 4: First side-bar prototype.

PageLinker 0.3 presents links to the left of the main window.

EVALUATION

Evaluating history-based tools such as PageLinker poses interesting methodological challenges with respect to validity [10]. We considered the following possibilities:

1. A **laboratory experiment** is easiest to control but poses external validity problems. Our fieldwork indicated that biologists' navigation and bookmarking behavior on unfamiliar tasks with artificial data might differ greatly from their behavior with familiar data and resources, making the results potentially meaningless. Also, users cannot fully leverage their personal knowledge in a lab experiment nor take advantage of their episodic memory. We are also interested in gathering realistic adoption and

Ver	Link type	Creation	Access
0.1	directed, not suppressible	Cut/copy paste	Menubar popup
0.2	bidirectional, suppressible	Cut/copy paste Menubar popup (open & last 7 pages)	Menubar popup
0.3	bidirectional, suppressible, always visible	copy/cut-paste list last-visited pages in Bookmark sidebar	Bookmark list via sidebar
1.0	bidirectional, suppressible, always visible	copy/cut-paste popup via Bookmarks sidebar shows last visited pages	Bookmark list via sidebar

Table 1: Four versions of PageLinker .

usage data for PageLinker: not only measuring performance advantages, if any, but also observing how user behavior evolves over time and whether users make the tool part of their repertoire.

2. An **uncontrolled field study** has greater external validity but is very difficult to control. Longitudinal field studies require extensive logging and extensive data analysis, especially if the participants' environment is not modified. Long-term monitoring also raises serious privacy issues and risks interfering with biologists' confidentiality agreements. For example, some biologists asked us to stop recording during the interview if they thought we might see confidential data. These biologists would not have been willing to participate in long-term automatic recording of their activities. Biologists also alternate between periods of intense on-line data analysis and periods of laboratory research. At any point in time, individuals may be out of phase with each other, depending upon who is writing a paper, running an experiment, or analysing data. This diversity complicates any comparisons and analysis of activity logs. For example, it would be difficult to tell, for any one subject, whether a decrease in pages visited was due to PageLinker or an overall change in research activity. It would also be difficult to compare people who were at different phases in their work.
3. A **limited time-series field experiment (or quasi-experiment [10])** offers the optimal compromise, with the external validity of a field study and most of the control offered by a laboratory experiment. Because we wish to compare PageLinker's navigation performance to existing browsers, it makes sense to alternate PageLinker with the user's usual browser. This allows us to track changes in use over time, based on realistic tasks performed in the user's real work setting, together with their existing bookmarks and other revisitation techniques. We chose this third option to evaluate PageLinker.

Method

Participants

Twelve biologists or bioinformaticians (9 men and 3 women between 20 and 40 years old) working in four research institutes (Institut Pasteur, Génopole, Université Paris 5, INRA) participated in the study. All were Firefox users with browsing and bookmarking experience. Two had also participated during the participatory design phase. (Post hoc analysis did not show significant differences between their results and those of other study participants.)

Apparatus

Hardware: Participants used their usual browser with their own bookmarks and history, on their own system: 5 Mac OS X users, 4 Windows users and 3 Linux users.

Software and logging: We used PageLinker 0.3 in the experiment and Navtracer³ [21], a standard Firefox extension that logs user interactions with the browser, to record user activity in both conditions. Navtracer runs on any version of Firefox (from 1.0 to 2.0) and could be installed and disabled rapidly in each user's browser without requiring special knowledge. This allowed us to minimise disruptions and let participants continue using their standard bookmarks, history and other Firefox extensions in both conditions in the evaluation.

To protect privacy, the extension does not begin logging automatically. Rather, users press a *start* button added to a Firefox window and fill out a form describing the experimental condition. This gives users full control of logging: they can pause, resume or stop at any time. When Navtracer was first installed, we showed users how to enable and disable logging and where the CSV log file was stored. They were invited to delete the file or modify its contents if they had concerns about what had been logged.

The extension registers various event handlers to detect the opening or closing of tabs and windows and the acquiring or loss of focus. It also tracks web-page changes and the relations between them via the page referrer. Switches between documents (windows or tabs) are also recorded. Event handlers append log data to a plain text file stored in the user's profile folder. Timestamps are systematically added to every record. Navtracer also logged PageLinker events such as link creation and usage of created links.

Scenario design

The experiment scenario was based on our observations of common tasks and navigation patterns, including:

- *Search:* web search engines, biological databases, directories
- *Parallel exploration:* same analysis with two programs
- *Results comparison:* same analysis with two programs
- *Analysis:* visual scan of results to check validity and pertinence

³ <http://navtracer.mozdev.org/>

- *Biological links directory:* scanning for options
- *Repeated path:* access the same page

We created a scenario with five related subtasks (Figure 5) with the aid of two biologists from the same environment. The scenario had to be short enough (between 15 and 20 minutes) so that it would not be too time consuming for participants, but still be representative of their tasks and understandable for every specialty. The five tasks illustrate aspects of web navigation presented above. The scenario is open and participants were encouraged to use their usual websites to perform the tasks. The websites presented here were the most commonly used, taken from different servers to illustrate the resource diversity faced by biologists. The five tasks are:

1: Database search: Find the gene corresponding to *human muscular dystrophy* and choose the nucleotide sequence attached to the TRIM32 gene (usually used NCBI⁴).

2,3: Parallel exploration: Analyse the nucleotidic sequence with two different tools, e.g., Genscan⁵ and Genemark⁶, to predict what the peptide sequence would be.

4: Comparison: Compare the two predicted sequences, e.g., using bl2seq⁷ to check if predictions are reliable (result R1).

5: Analysis and visual scan: Analyse one of the predicted peptide sequences to find regions of local similarity with other sequences with Blast⁸ (result R2). The goal is to find species other than homo sapiens that express the same protein with a high degree of confidence and are interesting for researchers looking for a related analysis or literature.

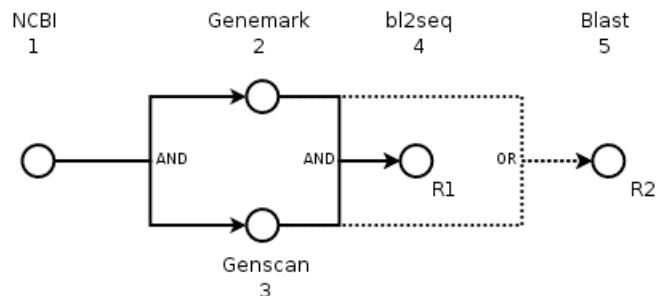


Figure 5: Scenario structure: Task 1 is performed first, followed by tasks 2 and 3 which are often performed in parallel. Task 4 is possible only after tasks 1-3 are complete and produces R1. Task 5 may be conducted independently after tasks 2 or 3 and produces R2.

Procedure

We used an ABAB within-subjects design, with one factor:

Firefox: Firefox browser with logging

PageLinker: Firefox browser with logging and PageLinker

⁴ <http://www.ncbi.nlm.nih.gov/>

⁵ <http://genes.mit.edu/GENSCAN.html>

⁶ <http://exon.gatech.edu/GeneMark>

⁷ <http://bioweb.pasteur.fr/seqanal/interfaces/bl2seq>

⁸ <http://www.ncbi.nlm.nih.gov/BLAST/Blast.cgi>

Users alternated between the PageLinker and the unmodified Firefox conditions at one-week intervals. Users kept their history, standard bookmarks and other Firefox extensions when changing conditions. This allowed them to work with their own real data settings instead of an empty initialized browser or one with artificial bookmarks and history the user was not familiar with.

Our goal was to collect data over long periods without extensive logging, so we sampled their navigation by taking a snapshot of the state of their bookmarks and asking them to follow the five-task scenario described above. Full counterbalancing of tasks across subjects is impossible, because PageLinker requires a first visit to websites to create the contextual links. (In other words, the unmodified Firefox condition must be run first, for all subjects.) We used an ABAB procedure, repeating each condition twice, to dissociate learning effects as much as possible from improvements due to PageLinker.

During the evaluation, each session was separated from the next by an interval of at least a week. Based on our previous observations, it appeared that seven days, including a week-end, should be long enough for participants to partially forget the exact details of what they had done during the previous session. This reduced the learning effect and is also representative of biologist's typical behavior: They frequently perform a series of tasks for one purpose and then repeat it after days or weeks of performing other tasks.

One experimenter visited each of the participants in their lab once a week for a month. During each visit, participants were asked to perform the same scenario. In the first session, we introduced PageLinker and invited the biologists to use it freely until they felt comfortable with link creation and use. This training period lasted between 10 and 15 minutes. We first showed participants how to create links either by copy/paste or the menu list. They were then free to try creating lists between any pages they liked. We finally asked them to determine pages they thought were related to each other and to create links between them using the two techniques. In case they had no idea of what to link, we suggested that they create links between pages they had visited during a recent break so as to avoid conflicts with our scenario. (Note: This occurred primarily during the first session, with a few biologists who had not done this type of analysis for a long time.)

The experimenter then presented the standard scenario, explaining its biological purpose and the necessary steps to achieve it. During this phase, we avoided mentioning any particular online tools and encouraged participants to use their favorite applications, portals or search engines. Our only guidance consisted of reminding them of the next task after they completed the previous one. Tools and portals were only suggested if they did not know what software was appropriate for a task or if their usual application server was down. (Note: The server went down twice in the course

of the month-long study and ran very slowly approximately once per participant.)

The PageLinker extension remained installed during all phases of the study, but was invisible to users during the Firefox-only conditions. In the latter case, it simply logged the creation of links between pages via copy/paste, as a conventional history tool. To protect privacy, we disabled the logging extension after each session. We also asked users if they wanted PageLinker to be disabled between sessions: All decided to keep it. To avoid interference between contextual bookmarks created during the experiment and non-experiment phases, we stored the contextual bookmarks in different files.

Predictions and Hypothesis

Based on feedback from our first field release and our personal use of the extension, we predicted the following results: We predicted that PageLinker would generate fewer pageloads and fewer clicks per task and reduce time spent on each task. We also predicted that with PageLinker, the majority of links would be created on the first visit to each relevant website. Since we had interacted with the users and iteratively responded to their requests during the design of the tool, we also expected our participants to be mostly satisfied with the design and interaction techniques used in the main experiment.

RESULTS AND DISCUSSION

Quantitative Results

PageLinker performed significantly better than the unmodified Firefox browser with respect to the following dependent variables (Figure 6):

- task completion time was 28% shorter ($p < 0.01$)
- 22% fewer clicks occurred ($p < 0.01$)
- 38% fewer pages loaded ($p < 0.01$)

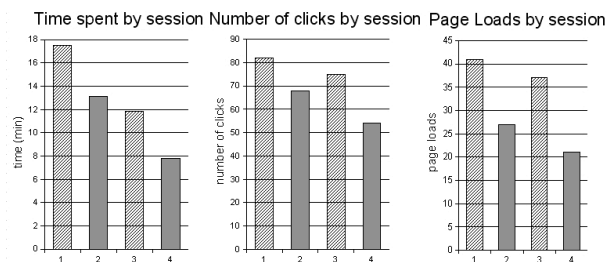


Figure 6: Evolution of time, clicks and page loads over sessions. Columns 1 & 3 are Firefox only, columns 2 & 4 are PageLinker.

If we focus more specifically on the limited time series, we observe the same pattern for clicks and page loads, although the difference is only significant for the number of page loads. The decreased number of page loads corresponds to the biologist seeing 38% ($p < 0.05$) fewer pages during a typical day. Although there is an overall learning effect, i.e. biologists become more efficient running the tasks in the scenario over time, there is also a strong effect of

PageLinker. Columns two and four (PageLinker conditions) are always more efficient than columns one and three (Firefox-only conditions).

The overall number of links created is not significantly different over the four sessions: A mean of 20 contextual bookmarks were created during the first session and 12 during each other session. Participants never had too many contextual bookmarks, with the corresponding risk of overload. This is because the use of contextual bookmarks increases linearly with the number of created links $F_{1,11} = 8.73$, ($p < 0.05$). In summary, these results suggest that PageLinker actively facilitates page revisitation:

- Fewer page loads shows that users visited fewer search websites and transition pages,
- Fewer clicks shows they used fewer transition pages, and
- Fewer pages seen shows they took less time to complete the five tasks of the scenario.

Qualitative Results

The participants' use patterns in the Firefox-only condition were very similar to those we saw in the earlier design phase. For example, they used directories of biological resources to find links to on-line programs and said that they usually preferred to use search engines to find a link, even when they knew that they had a bookmark for that particular page.

We observed several ways that PageLinker assisted users in their work flow. When interruptions occurred during the evaluation, such as people asking questions, coffee breaks, and phone calls, PageLinker helped them reorient themselves when they returned to their task. By seeing the links to and from the pages, participants could more easily remember what they were doing and what their goals had been. We also observed that it helped users in case of server slowdown or breakdown. They began to keep alternate links to the same program on different servers, something they never did with standard bookmarks because it would have generated an unacceptably large number of bookmarks. Unlike automation tools, PageLinker is robust to changes in internal page structure. The simplicity of our solution allows easy re-linking whenever a website's structure changes.

Limitations of the Experiment

Dissociating PageLinker effects from learning effects is complex when interpreting the time spent on the scenario and the number of clicks. Time is highly correlated with external factors, such as the current server load. For example, users may wait more than five minutes for a Blast result from the NCBI if the servers are heavily loaded.

Another potential problem is assessing the correlation between the number of contextual bookmark links and their use. Perhaps a month-long evaluation is too short to overload the contextual bookmarks menu. We expect that the recency classification we use, which only shows the

most recently used links, should reduce the overload effect, but we would need a much longer study to find out.

Finally, PageLinker can only reduce hyperlinks clicks, not the clicks needed to fill in forms. Nevertheless, the logger counted all clicks indiscriminately, whether they occurred on links or on forms. PageLinker thus accounted for only a small percentage of the overall number of clicks and the reduction was indistinguishable from noise.

Three-Month Follow-up

After the evaluation, we released PageLinker 1.0 which modified how contextual bookmarks are created. Figure 7 shows that the *link to* list has been changed into a menu.

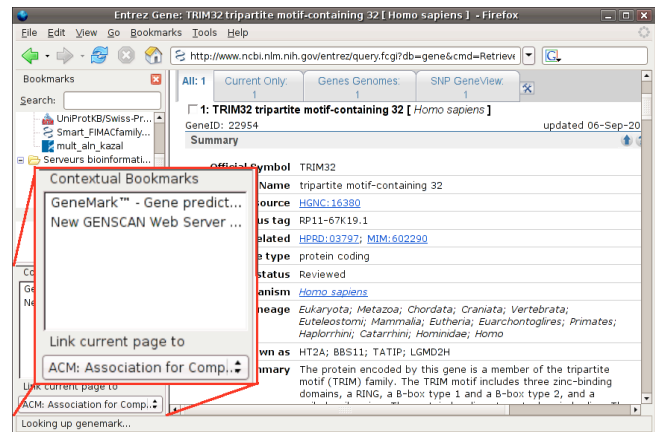


Figure 7: PageLinker version 1.0.

Three months later, we sent the participants a questionnaire (Table 2). Of the 12 participants, two had changed institution and did not answer, two had changed browser or workstation without re-installing PageLinker and eight still used PageLinker. The key questions in the questionnaire are presented in Table 2:

Question	Mean	SD
How usable is the link creation?	4.33	0.87
How usable are the created links?	4.44	0.73
How useful are the links created?	3.56	1.24

Table 2: Responses to the questionnaire using a five point Lickert scale: 1 = *not at all*, 5 = *very*.

Participants reported two primary uses of PageLinker in the months following the field experiment. The first is similar to that described in our scenario, in which users create chains of web pages, applying results from one page to subsequent forms. The second, more frequent PageLinker use involves creating relations between web pages that the users navigate frequently. We call this fuzzy grouping: the pages are related to each other without the hierarchical order imposed with regular bookmarks or other link organisers. If the user's area of interest changes slightly and they visit new sites, they simply add a few links to the

pages already linked to them and forget about obsolete links.

CONCLUSION

We began by addressing a specific problem faced by biologists: linking data output pages to data analysis pages. After we released the first version of PageLinker, users appropriated it, thus revealing the need for a more general contextual bookmark tool. Users sought ways to associate pairs of web pages and thus facilitate future navigation within groups of previously visited pages. Our studies with biologists demonstrated that PageLinker's contextual bookmarks improve web page revisitation and that, unlike history and bookmarks mechanisms, they are less prone to information overload. The philosophy of letting users handle their links allowed the tool to be both simple and robust to changes in Web content and user practices. Three months after the study, at least two-thirds of the participants were still using PageLinker.

Contextual bookmarks display links to other pages relevant to the user, depending upon the web page visited. Study participants considered these links both easy to create and to use. They also found them easy to understand and predict, since relevance is not decided by an automatic process but by users' explicit actions.

Future work includes the development of visual cues to indicate where users are in their navigation, presenting previously seen pages before and after the one being displayed on screen. Users also expressed the desire to share their links with others in their research team.

Biologists are heavy users of web browsers and are thus a good target audience to study when exploring navigation problems. Yet, the concepts developed for PageLinker are more general and likely to apply to a wide variety of users. In his classic article, "As We May Think", Vannevar Bush [4] argues that the human mind operates by association, connecting items into a web of trails. In the spirit of his Memex idea, we offer a tool that allows users to "build a trail of interest through the maze of materials available". Linking web resources while navigating is a powerful way to find information again and to reflect the users' thinking as they explore.

ACKNOWLEDGMENTS

We would like to thank all the biologists who volunteered their time for interviews, workshops and the study. Special thanks to Sophie Créno and Florence Hantraye in particular, for their help creating the scenarios, as well as Samira Laribi and Bertrand Neron for their generous explanations of biological research and processes. Partial funding was obtained from the MicroMégas project.

PageLinker is available at: <http://rabidfox.mozdev.org>

REFERENCES

1. D. Abrams, R. Baecker, and M. Chignell. Information archiving with bookmarks: personal web space construction and organization. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 41-48, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.
2. V. Anupam, J. Freire, B. Kumar, and D. Lieuwen. Automating Web navigation with the WebVCR. *Computer Networks*, 1:503-517, 2000.
3. S. Baker. Mosaic-surfing at home and abroad. In *SIGUCCS '94: Proceedings of the 22nd annual ACM SIGUCCS conference on User services*, pages 159-163, New York, NY, USA, 1994. ACM Press.
4. V. Bush. As we may think. *Atlantic Monthly*, 176(1):101-108, June 1945.
5. T. B. C. Sorensen, D. Macklin. Navigating the world wide web: Bookmark maintenance architectures. *Interacting with Computers*, 13:375-400, 2001.
6. L. D. Catledge and J. E. Pitkow. Characterizing browsing strategies in the world-wide web. In *Proceedings of the Third International World-Wide Web conference on Technology, tools and applications*, pages 1065-1073, 1995. Elsevier North-Holland, Inc.
7. A. Cockburn, S. Greenberg, S. Jones, B. McKenzie, and M. Moyle. Improving web page revisitation: Analysis design and evaluation. *IT Society*, 1:159-183, 2003.
8. A. Cockburn, S. Greenberg, B. McKenzie, M. Smith and S. Kaasten. *WebView: A graphical aid for revisiting Web pages*. Proc OZCHI'99, Australia, 1999.
9. A. Cockburn and B. J. McKenzie. What do web users do? an empirical analysis of web use. *Int. J. Hum.-Comput. Stud.*, 54(6):903-922, 2001.
10. T. Cook and D. Campbell. *QuasiExperimentation: Design & Analysis Issues for Field Settings*. Houghton Mifflin Company, 1979.
11. J. Fujima, A. Lunzer, K. Hornbaek, and Y. Tanaka. Clip, connect, clone: combining application elements to build custom interfaces for information access. In *UIST '04: Proceedings of the 17th annual ACM symposium on User interface software and technology*, 2004.
12. R. Hightower, L. Ring, J. Helfman, B. Bederson, and J. Hollan. Graphical multiscale web histories: A study of PadPrints. In *Hypertext '98 Proceedings*, pages 58-65, 1998.
13. W. Jones, H. Bruce, and S. Dumais. Keeping found things found on the web. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 119-126, New York, NY, USA, 2001. ACM Press.

14. C. Letondal. A web interface generator for molecular biology programs in unix. *Bioinformatics*, 17, 2001.
15. C. Letondal and W. E. Mackay. Participatory programming and the scope of mutual responsibility: balancing scientific, design and software commitment. In *PDC 04: Proceedings of the eighth conference on Participatory design*, pages 31–41, New York, NY, USA, 2004. ACM Press.
16. W. Mackay, C. Letondal, G. Pothier, K. Bøegh, and H. Sørensen. The missing link: augmenting biology laboratory notebooks. In *UIST 2002*.
17. W. Mackay. *Users and Customizable Software: A Co-Adaptive Phenomenon*. PhD thesis, M.I.T., 1990.
18. W. Mackay. Triggers and barriers to customizing software. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 153–160, New York, NY, USA, 1991. ACM Press.
19. W. Mackay and A. L. Fayard. Video brainstorming and prototyping: techniques for participatory design. In *CHI '99: CHI '99 extended abstracts on Human factors in computing systems*, pages 118–119, New York, NY, USA, 1999. ACM Press.
20. J. Nielsen. User interface directions for the web. *Commun. ACM*, 42(1):65–72, 1999.
21. N. Roussel, A. Tabard, and C. Letondal. All you need is log. In *WWW 2006 Workshop on Logging Traces of Web Activity: The Mechanics of Data Collection*, May 2006. 4 pages.
22. Stevens, Goble, Baker, and Brass. A classification of tasks in bioinformatics. *BIOINF: Bioinformatics*, 17, 2001.
23. Suchman, L. (1987) *Plans and Situated Actions*. Cambridge, England: Cambridge University Press.
24. L. Tauscher and S. Greenberg. How people revisit web pages: empirical findings and implications for the design of history systems. *International Journal of Human Computer Studies*, 47(1):97–137, 1997.
25. L. Tauscher and S. Greenberg. Revisitation patterns in World Wide Web navigation. In *Proceedings of the Conference on Human Factors in Computing Systems CHI'97*, 1997.
26. J. Teevan, C. Alvarado, M. Ackerman, and D. Karger. The Perfect Search Engine Is Not Enough: A Study of Orienteering Behavior in Directed Search. In *Proceedings of ACM CHI 2004 Conference on Human factors and Computing Systems*, pages 415–422. ACM Press, Apr. 2004.
27. H. Weinreich, H. Obendorf, E. Herder, and M. Mayer. Off the beaten tracks: exploring three aspects of web navigation. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 133–142, New York, NY, USA, 2006. ACM Press.