A day in the life of ten WWW users

Michael D. Byrne

Department of Psychology

Rice University

Houston TX

byrne@rice.edu

Bonnie E. John

Human-Computer Interaction Institute

and the Departments of Computer Science and Psychology

Carnegie Mellon University

Pittsburgh PA

bej@cs.cmu.edu

Elizabeth Joyce

Department of English & Theater Arts

Edinboro University of Pennsylvania

Edinboro, PA

ejoice@erienet.com

Abstract

We videotaped ten volunteers as they used the World-Wide Web during a normal work day, asking them to think-aloud as they used the Web. We analyzed these videotapes at the level of user-intentions to form a taxonomy of tasks these people performed. We also coded these tapes at the level of user actions. The data reveal that several previous claims about browsing behavior are questionable, and suggests that that widget-centered approaches to interface design and evaluation may be incomplete with respect to good user interfaces for the Web.

Keywords

World-Wide Web, task analysis, video protocols

1 Introduction

A great deal of public and research interest has been devoted to the World-Wide Web (or WWW) in recent years (see, for example, the Human Factors and the Web conference series (http://www.itl.nist.gov/iaui/vvrg/hfweb/conferences/prev_conferences.en.html). Most of this research effort has focused on the technical aspects of the Web and the application of Web-based technologies (e.g., an analysis of the BACK button, Greenberg & Cockburn, 1999; introduction of Web Macros, Safonov, Konstan, & Carlis, 1999). Our concern, however, is with the usability of the Web. In order to perform any kind of sensible usability evaluation, be it empirical, analytical, or heuristic, it is necessary to first understand the tasks users engage in while browsing. That is, in order to determine which HCI techniques and/or approaches are most likely to aid in Web usability, it is first necessary to understand what is that users actually do with their time while using the Web.

There is an abundance of marketing data on how people use the web, measured with a variety of methods. For instance, Media Metrix has a “large, representative sample of over 50,000 individuals under measurement at home and at work” who supply demographic information as well as monitored performance (http://www.mediametrix.com/). Cyber Dialog (http://www.cyberdialogue.com/) screened 13,000 people to interview 1000 on-line uses and 1000 non-users with a 150+ question interview; they also conduct focus groups on internet use. Although these data are valuable to the business community for marketing strategies and site design, they are unavailable freely to the research community. Some research studies are available, e.g., Catledge & Pitkow, (1995), Tauscher & Greenberg (1997), but in a much more limited number. More important than availability, though, is that these studies do not necessarily tell us what we need to know for web system design. Interview and focus group data do not provide performance in context, only a person's memory or conjecture about his or her performance. On the other hand, "click studies" (where the data are the links the user clicked on, or pages they visited) provide little information about the task contexts in which the user's actions occurred. For example, a click study can provide information about how often pages are visited and links are traversed, but not the tasks in which users were engaged while doing so. The focus of the present research is to gain a clearer understanding of the tasks users engage in while browsing the Web and the time spent doing those tasks. This will enable rating the relative importance of various interface analysis methods—there is little to be gained by analyzing tasks that users rarely perform, or that cost users very little time.

Further, we wanted to observe the tasks users normally perform in their daily Web use, rather than giving them artificial tasks as some studies have done (e.g., Morkes & Nielsen, 1997; Nielsen, 1994; Lynch. Palmiter & Tilt, 1999; Navarro-Prieto, R., Scaife, M., & Rogers, Y, 1999). While observing users doing specific tasks can be useful, it is possible that the task or tasks used in the study might not reflect the tasks that users do when left to make their own decisions about how their time is allocated. Other naturalistic studies of user behavior not on the Web (e.g., Cypher, 1986) have found that undirected user behavior is much more complex and interleaved than directed behavior.

The next section gives the details of the method we used to collect our naturalistic data. Section 3 details our analysis process. Section 4 presents and discusses our results. Section 5 explorers their implications for the design and evaluation of WWW-related interfaces.

2 Method

2.1 Participants

Participants were 10 members of the Carnegie Mellon University community who volunteered to participate in the study. They were all experienced Web users, and represented a cross-section of users as faculty, students, secretarial staff, and research staff were included. Specifically, we had one faculty member, one member of the research staff, one member of the secretarial staff, and seven student volunteers.

2.2 Procedure

Our goal was to collect a "day in the life" of each user, during which they would browse the Web as they would in a normal day. In order to encourage the participants to engage in the kind of browsing they would do normally, they were videotaped in their offices (or home in the case of one student participant) using their normal workstation.

Each participant was asked to participate for one day of the study. On the morning of that day, a video camera was set up in the participant's office (or home in the case of one student participant) to record both the user's screen and their protocol. The participant was trained in how to use the camera and given instructions about providing verbal protocols. Extra blank tapes were also left with the participant in the event that recording exceeded the capacity of a single tape. The camera and any tapes were collected at the end of the day.

Participants were asked to work normally as they would on any other business day, but to turn on the video camera and provide protocol whenever they used the WWW. While it is possible that participants altered their browsing patterns due to the presence of the video camera, we explicitly discouraged this. In fact, we had one user who returned a blank tape to us because they simply didn't browse the Web at all that day.

2.3 Materials

A variety of platforms and browsers were used, with Netscape Navigator running on a Power Macintosh being the most frequent choice. Almost seven hours of videotape was analyzed, of which just under six hours (5:55:29) was WWW browsing. The amount of data generated and analyzed for each participant ranged from approximately 17 minutes to just under 50 minutes. One of three analysts coded each videotape. Each participant was recorded with a Hi8 camcorder. The Hi8 tapes were either analyzed directly or digitized into MPEG files and put on CD-ROMs. The video and MPEGs were controlled and coded with MacSHAPA software (Sanderson, et. al. 1994).

3 Analysis

The videotaped protocols were coded at two levels: basic actions and tasks. The task-level coding is more central and will be described first. We constructed a taxonomy of tasks (a "taskonomy") to describe the kinds of tasks encountered in WWW browsing behavior. The taskonomy was originally developed through a combination of our prior research into information-finding (Peck & John, 1992), analysis of the the capabilities of the Web and Web browsers, personal introspection, and observing our first three users. This generated the taskonomy that was presented in Byrne, et al. (1999). The taskonomy presented here represents a refinement of that taskonomy based on comments and discussions with other researchers regarding the original taskonomy.

We classified the tasks into eight categories: Use Information, Locate Information, Provide Information, Find on Page, Navigate, Configure Browser, Manage Window, and React to Environment.

As an example of how we constructed this taskonomy, consider the task Locate on Page. Our prior research in the use of a textual on-line help system (Peck & John, 1992) indicated that when attempting to locate information, users could search for a specific string or a related concept. Since web pages also include images, we included a category for locating an image. Introspecting, we added a category for locating something "interesting." Finally, we observed one of our first three users searching for information that would be tagged with a specific string (explained below).

We chose to code the protocols at this level because we believe this to be a useful first pass at understanding how users allocated their time in terms of tasks and behaviors. Other levels of abstraction are both possible and potentially useful and should be pursued in future research.

3.1 Use Information

Use Information describes any activity (or series of activities) in which the user was attempting to use a piece of information from the WWW. The Use Information subcategories were based on our observations of what the first three users did with the information they obtained from the Web. Information on the Web can serve a variety of purposes: it can be read, listened to, viewed or watched (e.g. images, animations, layouts), duplicated (e.g. copy and paste), downloaded to a local disk, displayed for other people to see, printed, etc. Most activities done while browsing the Web are in service of a Use Information task.

A Use Information task was coded as beginning whenever one of the following criteria were met:

- When the user said they were doing something new (i.e., verbalized a new goal) before they take action.
- When the user said they were doing something new (i.e., verbalized a new goal) simultaneously with taking action.
- When an action could be inferred to be the start of a new goal even though the verbalization of that goal was later (e.g., they clicked on a link and then, while waiting for the page to come up, they said why they did that or what they were hoping to find).
- When they had finished an old goal (e.g., said "ok" or something like that) AND then started doing something else even if they had not verbalized that new thing until after the action.

A Use Information task was terminated whenever any of these conditions were met:

- When the user explicitly said they were done.
- When the user implicitly "said" they were done, e.g., the user left the room.
- When the user started a new UseInfo as defined above.

We also coded whether Use Information tasks were successful or not. We defined three values for this: "success," "failure," and "changed." A Use Information task was considered a success when the user explicitly expressed satisfaction with what they had found or done, or when the goal was clearly achieved (e.g. when the user had said they wanted to read today's weather and today's weather was on the screen and they had read it).

A Use Information was considered a failure if any of the following conditions were met: they explicitly stated that it was a failure, they explicitly gave up, or the user switched goals without a successful end (as just defined).

A Use Information task was considered changed whenever the user changed the desired use. For example, if the user's original goal had been to read something on the Web, but upon finding it, they decided it was too long and to just print it instead, then this would be coded as changed.

3.2 Locate Information

A frequent precondition to doing something with a piece of information (a UseInfo) on the Web is getting the browser to that information. When users were engaged in this, they were coded as being in a Locate Information task. LocateInformation tasks were coded as starting with whatever action initiated the search, typically a mouse move towards some navigation functionality. The task was terminated as soon as an acceptable page became viewable.

We not only tracked the beginning and ending of such tasks, but something about their character as well. In particular, we were interested in three things: what kind of criteria were being used for the page being sought, what method users employed to reach web pages, and whether or not they were successful in doing so. We defined four types of searches: specific, related, tagged, and interesting. "Specific" searches occurred when the user was looking for a specific page. "Related" searches occurred when the user was looking for a page with information related to some topic, but not a specific page.

"Tagged" searches occurred when the user was looking for something specific, but not a specific location. For example, a user might have been looking for a page with the price of a new G3 PowerBook. Thus, pages just about PowerBooks in general (a related search) do not qualify, only one with this specific information on it. However, any page with this information will do, so this is not quite as exact as a "specific" search. Finally, there were "interesting" Locate Informations, which occurred when the user was simply looking for something interesting in some way.

We also defined three methods for initiating a Locate Information task. They are "search engine," "known start," and "unknown start." When the user made use of a search engine or searchable index, "search engine" is how it was coded. If the user started from a Web site that they clearly knew, then the Locate Information was coded with "known start." Known starts were coded when the user selected from a bookmark or followed links from their home page, or through use of the back/history mechanisms. Also, if they started by typing in a URL or an external launch (which is any launch not initiated in the browser itself, e.g. by clicking on a URL in a mail program), then they had to provide verbal evidence that they knew something about where that URL would take them for it to have been coded as a "known start." "Unknown starts" occurred when the user started by following links from a page with which they were not familiar, for example URLs received in chat rooms, or a guess of a university or company name, or a URL seen in an ad.

Success, failure, and change were also coded for Locate Information tasks in much the same manner as for Use Information tasks.

3.3 Provide Information

Users not only use the Web to get information, but to send it as well. They provide product selections, authentication information, shipping addresses, search criteria, and so on. These activities were all classified as Provide Info tasks. Provide info tasks were coded as beginning as soon as the user began the mouse move or typing that supplied the information (usually in a form) and ended as soon as there was confirmation that the information had been received (typically by the display of the response page). There are a potentially infinite number of kinds of information users could be providing, so we made no strong commitments to particular subcategorization.

3.4 Find On Page

Frequently, using a piece of information or going to a URL requires finding that information or link on a Web page, which typically requires some visual search. We called these activities Find On Page tasks. Users could search for a specific word, which we called Find On Page "string" searches. Users also searched for particular images (e.g. graphic links), coded as Find On Page "image." They could have been looking for something not necessarily a particular word or image but anything related to a concept (e.g. "I'm looking for 'photography' or 'cameras' or something like that"), which we termed Find On Page "related." Another class of searches can be best described as Find On Page "interesting," in which a user was seeking no specific word or concept, but is simply looking for something that might catch their interest. The most difficult kind of search to explain, but one which was observed, was what we called Find On Page "tagged." When a user was looking for a particular piece of information and did not know what it was that they were looking for, but knew some tag that would identify it as the piece of information they wanted, it was coded as Find On Page "tagged." For example, one user wanted to know the resolution of a printer he was considering purchasing. He did not know the number of dots per inch for the printer, but knew that the number he wanted would be tagged with something like "resolution" or "DPI" or the like. This is distinct from Find On Page "related" in that it is not the concept that the user is searching for, but a value pointed to by some tag matching a concept or word.

Find On Page tasks were coded as beginning either as soon as the relevant page was visually available to be searched (usually after loading) or the user's protocol gave evidence they were searching. A Find On Page was considered complete when the user explicitly indicated they had found the item or when a mouse movement was made to the target item. Alternately, a Find On Page task could be coded as finished when the user gave up, either explicitly or by navigating to a page not linked to the current one (e.g. clicking "back").

We also coded success/failure/change for Find On Page tasks as well, similar to how these were coded for Use Information and Locate Information.

3.5 Navigate

Any activity which caused the browser to display a URL different than the one currently being displayed we considered a Navigate. Most browsers support a wide array of ways in which a browser can be directed to a URL, including the back/forward button, bookmarks, hyperlinks, typing in a URL, history menus, a Home button, and others. Our subcategories of Navigate were based on an analysis of the methods supported by the browsers used. Navigate tasks were typically fairly rapid, but they could be timeconsuming, such as when typing a long URL was involved, the network response was slow, or the browser took a long time to render the page.

Navigates were coded as starting as soon as the command that caused the browser to change pages was initiated (for example, as soon as users pointed at the URL field in the toolbar) and were coded as complete as soon as the destination page was displayed with enough content that it was possible for the user to interact with it.

3.6 Configure

There is a wide variety of browser state information that is user-configurable, and changing the state of the browser (other than which URL to view) we termed Configure tasks. The kinds of Configure tasks available to the user depended on the number of user-configurable options provided by the browsing software. This was not a common task but there are things about the browser that users can change, such as bookmarks and assorted other preferences like cache size.

Configure tasks were coded as beginning as soon as a mouse moved or the keystroke involved in changing whatever aspect of the browser state change began, and ended whenever the final state at the end of the task had been reached.

3.7 Manage Window

This category is something like Configure, but in some ways more ephemeral. There are many aspects of window management to consider, such as moving, resizing, reordering, closing, and most commonly, scrolling. Like Configure tasks, Manage Window tasks were coded as beginning as soon as a mouse moved or the keystroke involved in changing whatever aspect of the window began, and ended whenever the final state of the window at the end of the task had been reached.

3.8 React

While most browsing activities are user-driven rather than browser-driven, there are times when the browser demands something of the user. We classified these situations as React tasks. These were typically in the form of a responding to a dialog box (e.g. where to save a file, can't find a DNS entry, etc.), but did take other forms. One common other form was the use of the Reload button-the user was reacting to some problem with a page display. Many React tasks had Manage Window tasks as subgoals. For example, when a page was loaded that has a fixed-width table in it that was wider than the current window, this often caused the user to react with a Manage Window task to change the window width.

React tasks were coded as beginning whenever a dialog or extraneous window appeared, or whenever the mouse movement to the control (e.g. the "Reload" button) required to react to the situation started. React tasks were considered complete when the dialog or window had been dismissed or when the action initiated by the React task completed (e.g. the page had reloaded).

3.9 Comparison to previous taskonomy (Byrne, et. al, 1999)

This taskonomy is similar to the one presented in Byrne, et al. (1999). Two categories have been renamed (GoTo has become Navigate and Locate has become Find On Page) and two categories added. One of the new categories is Manage Window, which used to be a subset of Configure. We split these out of Configure because they completely dominated the category and are in character somewhat different. In particular, window management will not affect the state of the browser in the next session, while changing preferences and bookmarks will. We decided that this justified splitting the category.

The other addition is more significant. This is the Locate Information task. Locate Information is something of an intermediate level between Use Information and Find On Page/Navigate. After carefully analyzing the the data in Byrne, et al. (1999), we concluded that the coding scheme used there, while all behaviors we saw fell into one of the coding categories, was missing an explicit representation of a very common task on the Web--finding the desired page. In some cases, a Locate Information will be uninteresting, consisting of a single Navigate. In other cases, it will be a complex interleaving of Navigates, Find On Pages, and even Provide Infos (e.g. when search engines are used).

3.10 Subtask Sequencing

In general, these tasks cascaded a great deal and had subtasks. For example, one user wanted to download a paper written by a colleague. Thus, the top-level goal was to Use Information (download). First, the user had to find that page on the Web, so a Locate Information task began (the Use Information task would still be active at this point). The user decided to use a search engine to find the colleague's page, which generated a Navigate (bookmark) task to get to the engine. Once there, the user engaged in a Provide Information task to tell the search engine what to look for, followed by a Find on Page task to find the appropriate link. This was followed by another Navigate (hyperlink) task to the relevant page, then another Find on Page to find a link to the paper itself. The entire episode counts as a single Use Information task, with several subtasks performed in a nested sequence:

Use Information (download, success)

Locate Information(search engine, tagged, success)

Navigate(bookmark)

Provide Information(search criterion)

Find On Page (related)

Navigate(hyperLink)

Find On Page (related)

This episode generated seven task instances. Note that the duration of the top-level Use Information task would include the time taken for all the subtasks--the task covers the time beginning when the user begins their attempt to download the file until the download is complete. Figure 1 presents the same information in a timeline format.

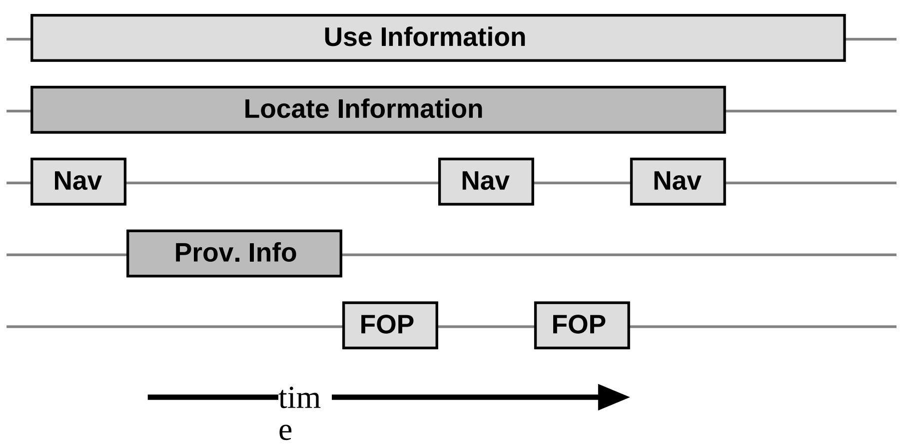

Figure 1. Timeline for nested subtasks with time represented on the horizontal axis. Nav = Navigate, FOP = Find On Page, Prov. Info. = Provide Information. <success> arguments for all categories have been omitted to conserve space.

Use Information tasks are not the only kinds of tasks that can have subtasks. In fact, all of the task types can (and did) have subtasks. Locate Information tasks typically included both Find On Pages and Navigates. Find On Page tasks often have Manage Window subtasks, such as scrolling the window. Provide Information tasks can generate Use Information tasks (often "duplicate") to provide form fill-in values. Configure tasks rarely had subtasks, but did occasionally (such as a Use Information subtask to determine what it is that a particular preference does). React tasks, as previously mentioned, often have Manage Window subtasks. Furthermore, tasks at any level could logically generate subtasks-this did not occur only at the top level. Since each task type can generate one or more of the other types as a subtask at any level, there is very little a priori hierarchy that can be imposed on the taxonomy.

4 Results and discussion

We discovered in earlier work (Byrne, et al. 1999) that each one of our task categories could generate any one of the other kinds of tasks as a subtask, and thus shied away from terming it a hierarchy. However, we have changed the taskonomy somewhat and believe that, while there are exceptions to this structure, there is in fact some hierarchy to the tasks. The organization that emerged from the data is presented in Figure 2.

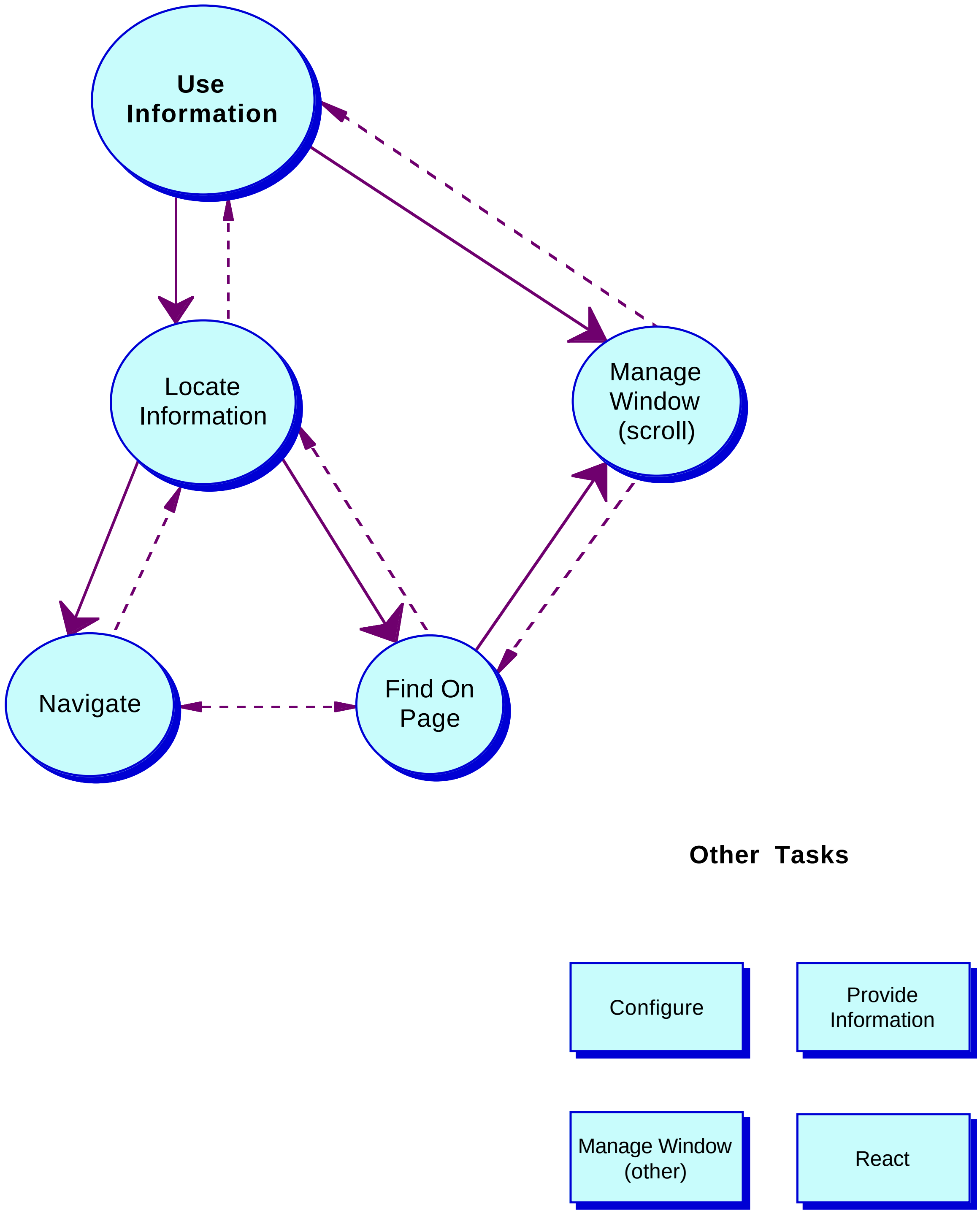

Figure 2. Taskonomy organization. Major tasks are indicated by circles, minor tasks by rectangles. Solid arrows with large arrowheads indicate transitions that were frequent and that indicate nesting, e.g. Locate Information nests inside Use Information. Dotted arrows with small arrowheads indicate transitions upon the termination of the task, e.g. when a Find On Page ended it typically transitioned to either a Navigate or a Locate Information.

Far and away the most common top-level task was Use Information. Each Use Information could generate zero to many Locate Information tasks, but most Use Information tasks generated one. Navigate and Find On Page tasks could occur outside the context of Locate Information, but they rarely did so. Within the context of a Locate Information task, users tended to oscillate between Navigate and Find On Page. Scrolling, which is denoted Manage Window (scroll) in the figure, happened primarily in two contexts, within Find On Page tasks and directly underneath Use Information tasks (typically after the user had located a page they wanted and they were reading it). Other tasks such as Configure and React could occur in any context, though they did not occur often in any context.

This decomposition is further supported by the number of events and total times observed. In 5 hours and 55 minutes of browsing, we found users in the context of Use Information for 5 hours and 51 minutes. They were in the context of a Locate Information task for 129.4 minutes, or over 2 hours. Total time in minutes in each of the major tasks is shown in Figure 3.

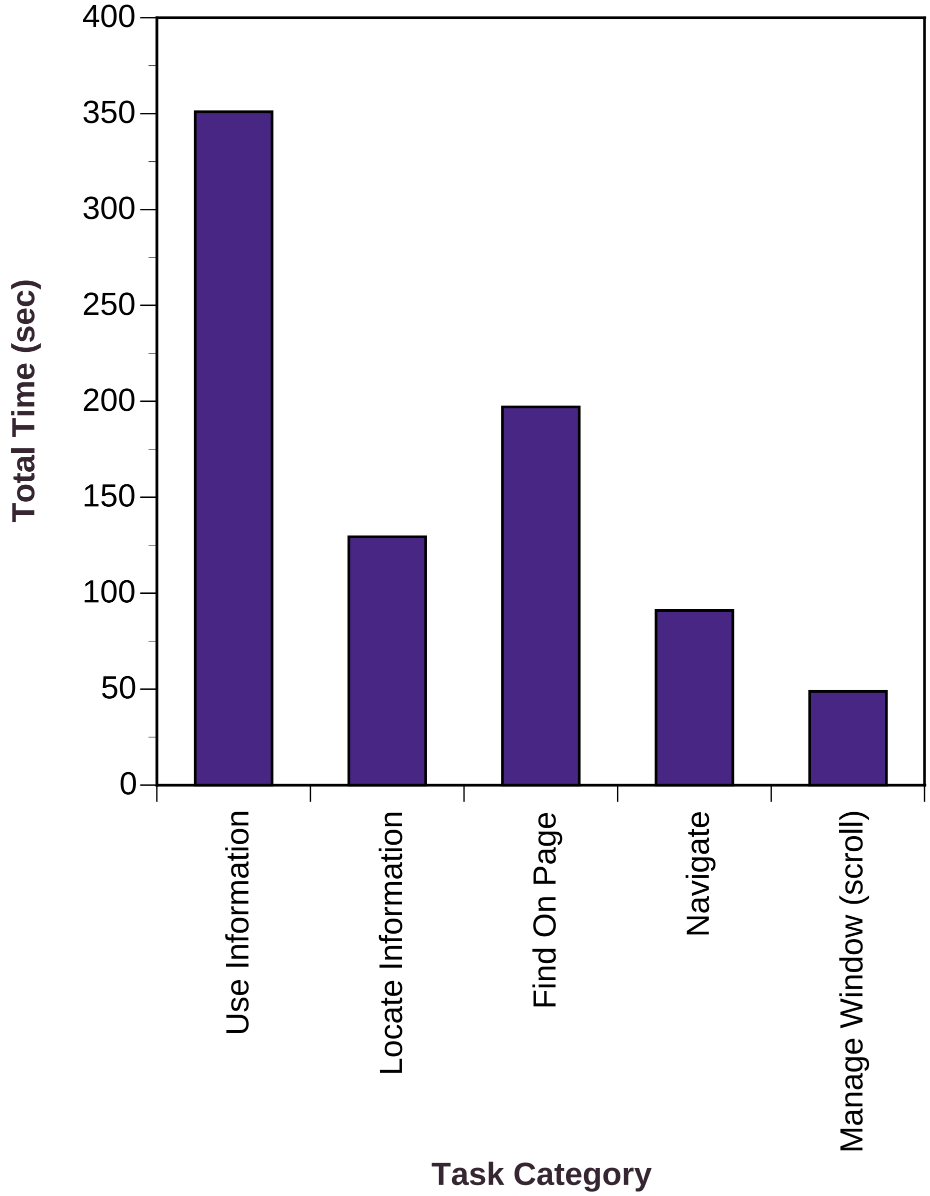

Figure 3. Total time for each major task category, in minutes.

Total number of tasks also conforms to this pattern. The number of Locate Informations was approximately equal to the number of Use Informations, but there were many more Navigates and Find On Pages. Further, the number of Navigate and Find On Page tasks was also similar, as these often happened in the context of Locate Information in pairs. Because both finding something on a page and reading a page for its content often required scrolling, a great deal of scrolling occurred. Number of events for each category are shown in Figure 4.

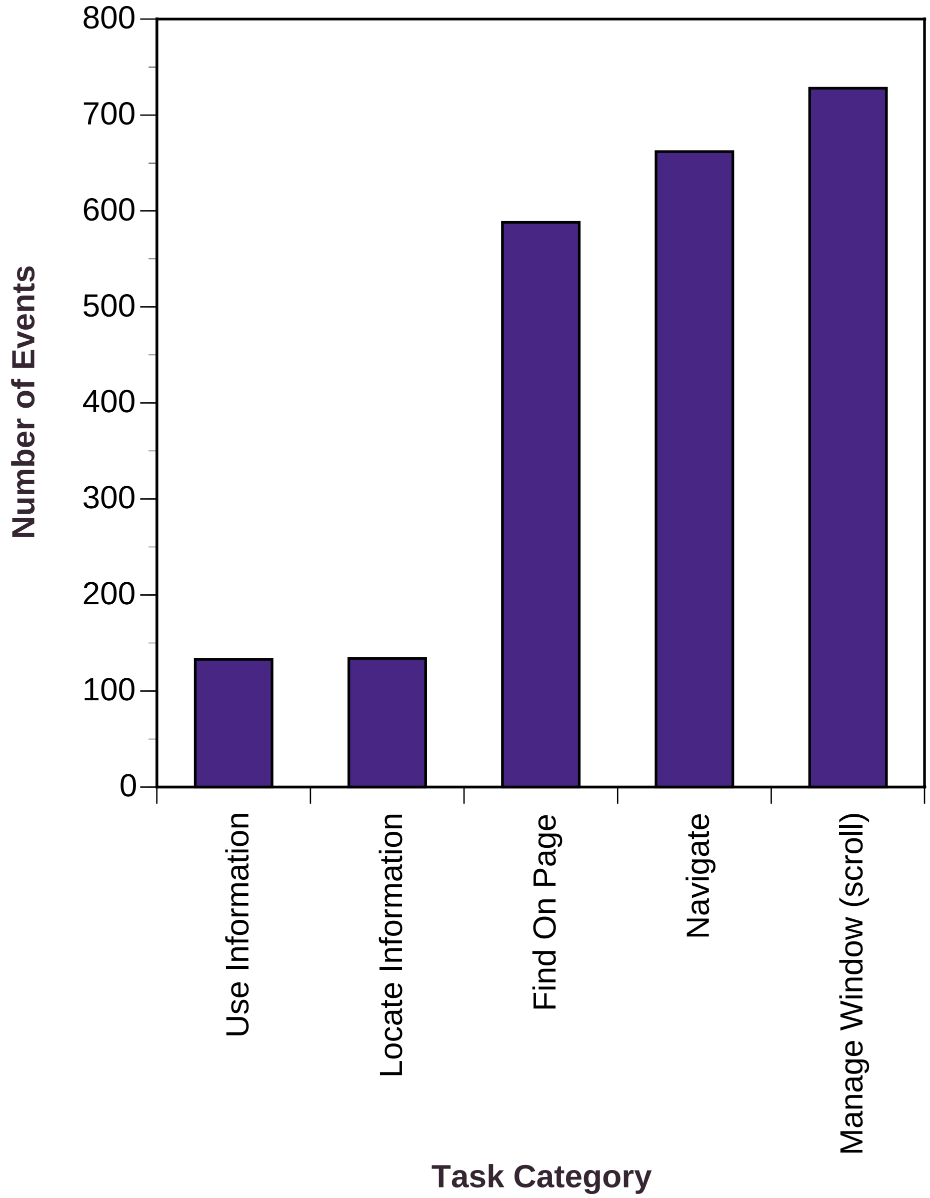

Figure 4. Number of events for each major task category.

Of course, most of these tasks had subtypes. We will describe the results for the individual subtypes in the following subsections.

4.1 Use Information

The task that most commonly served as the top-level task in our analysis was Use Information. We observed 133 Use Information episodes. We were unable to identify, either on the basis of the user's actions or their protocol, the type for six of these events so we eliminated them from consideration. The remaining 127 events had an average duration of 2.76 minutes each. That is, the average time our users spent on each piece of information, both finding it and making whatever use of it they made, was a little under three minutes. The range was substantial, from approximately two seconds to over 29 minutes, yielding a standard deviation of 4.42 minutes.

Use Informations were further classified by type (e.g. reading, downloading, printing) and outcome (success vs. failure). Despite the increase in multimedia content on the Web, the primary type of Use Information was clearly reading, which dominated all other alternatives combined by a factor of nearly four to one. This is in contrast to prior publications which claim that Web users did not read (Nielsen, 1994). We hypothesize that users read when they're doing their own tasks as opposed to those given to them by researchers. Figure 5 presents the frequencies of different Use Information tasks. However, there was no reliable difference in the mean lengths of the different types of Use Information tasks, $F(6, 125) = 0.98$, $p = 0.44$.

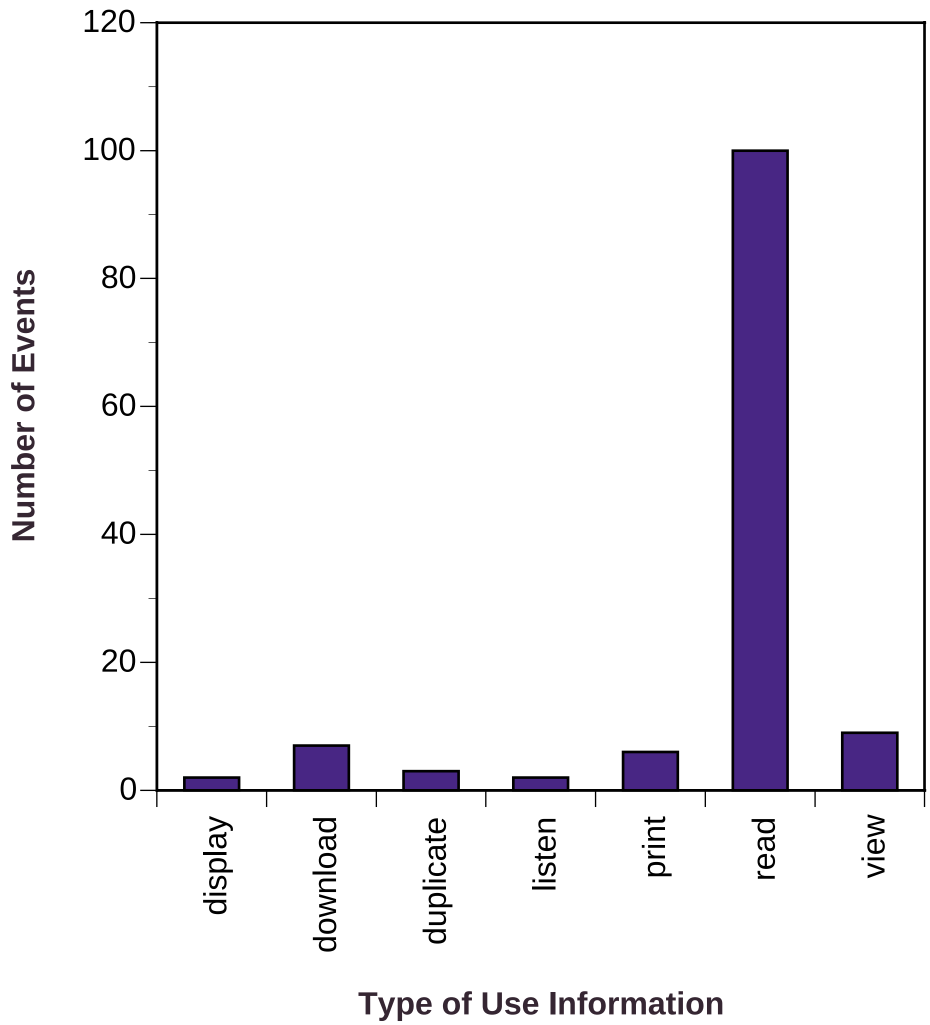

Figure 5. Number of events for each type of Use Information task.

By and large, users were successful in their attempts to use information on the Web. Of the 127 Use Information tasks, 102 were coded as successes, 22 as failures, and 3 as changed. There was again no evidence the the success of a Use Information task was related to its duration, $F(2, 124) = .40$, $p = 0.67$. Furthermore, there was no association between type of Use Information (e.g. reading vs. printing) and success, chi-square(12) = 4.14, $p = 0.98$. That is, users were no more or less likely to be successful if they were performing one kind of Use Information vs. another. A two-way analysis of duration by type and success was not possible because some of the cells were empty (for example, there were no Use Information tasks with a type of “printing” and an outcome of “changed”).

4.2 Locate Information

Before information could be used, our participants often had to make a effort to get to that information. We observed 134 tasks which we classified as Locate Information tasks, which were further subcategorized according to three factors: type, method, and outcome. Four of these could not be classified on at least one of the factors and will not be considered further. The remaining 130 Locate Information tasks took an average of 1.00 minutes, with a minimum of under one second and a maximum of 17.40 minutes. Each Locate Information task will be referred to as a *hunt*.

There were four different types of Locate Information tasks: interesting, related, tagged, and specific. Frequencies of the different types are presented in Figure 6. Related hunts were the most common, about twice as common as each of the other types. Mean durations for the different types of Locate Informations are presented in Figure 7. The average “tagged” hunt took about a minute and a half while the average “interesting” hunt took much less, under half a minute. However, the overall effect of type on the duration of Locate Information tasks was not statistically reliable ($F(3, 126) = 1.28$, $p = 0.29$).

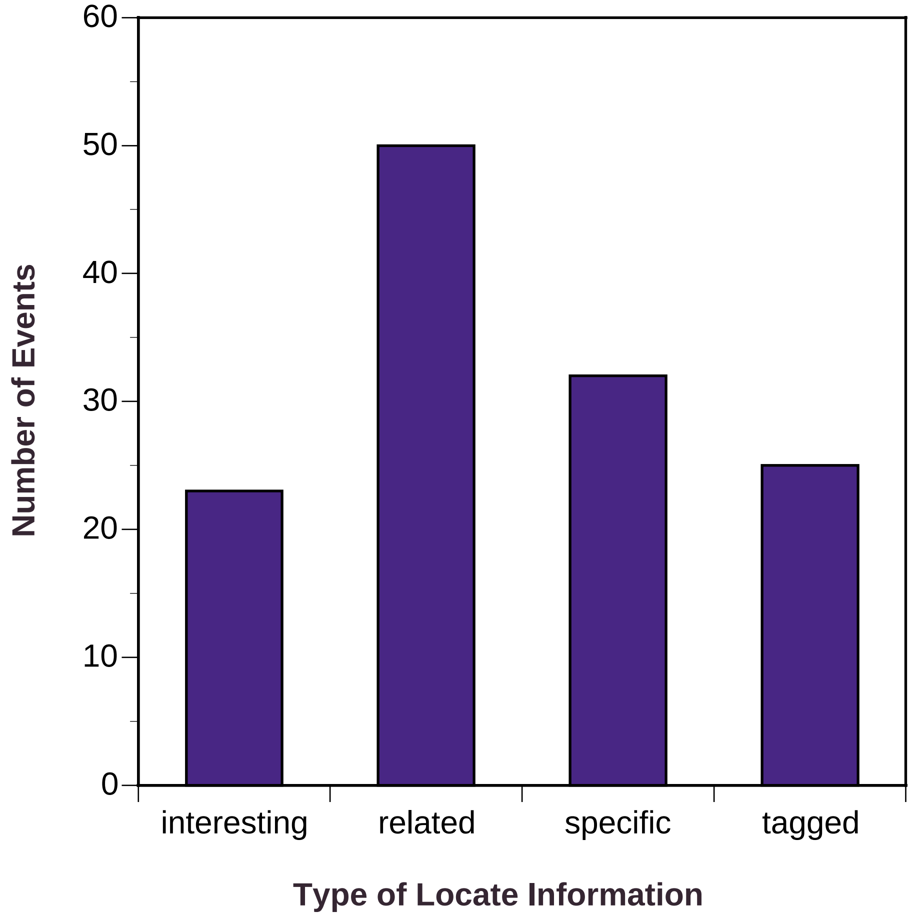

Figure 6. Number of events for each type of Locate Information task.

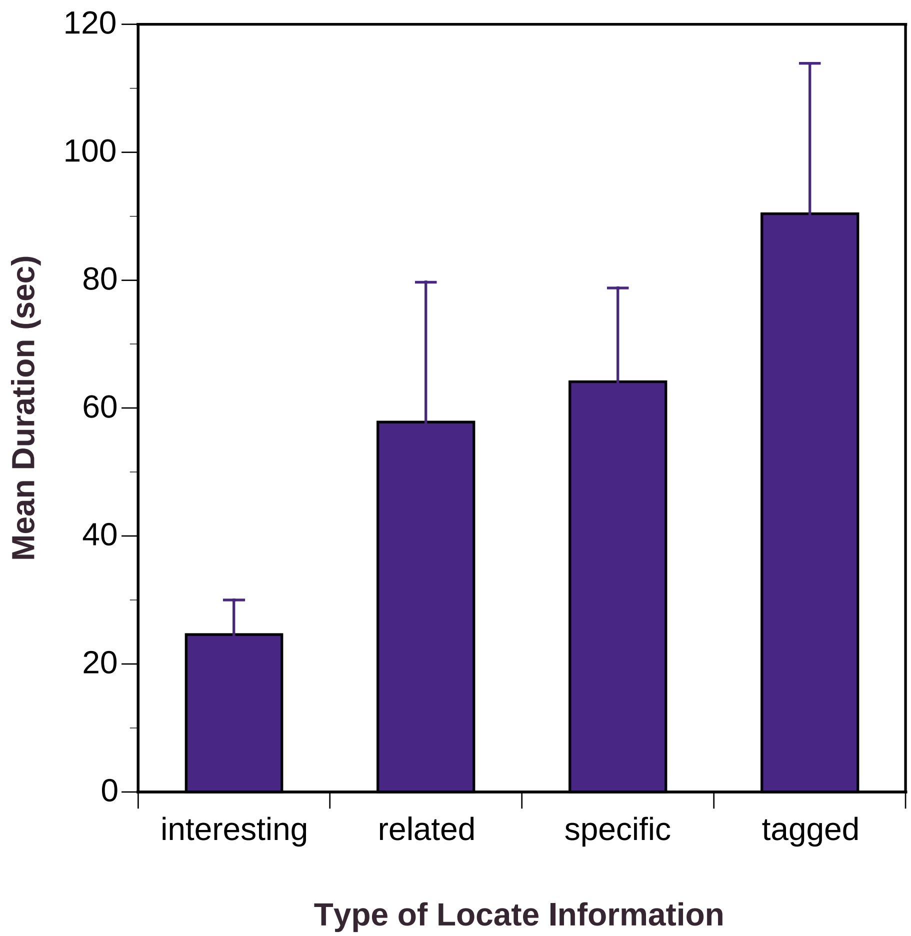

Figure 7. Mean duration for each type of Locate Information task. Error bars indicate one standard error of the mean.

Three different methods were identified: known start, unknown start, and search engine. Our users most often started from a known location (e.g. their home page, a bookmark, the last page visited), 73 times overall. 16 times search engines were employed, and 41 hunts were started at locations that were previously unknown to them. The method had a large impact on the time taken to locate the page containing the desired information ($F(2, 127) = 15.50$, $p < 0.001$), the average time for each method is presented in Figure 8. What is particularly interesting is that the use of a search engine is associated with markedly slower searches. This may be because search engines are used to find the information about which the user had the least idea where to look, or this may be because search engines simply are not efficient--these data do not allow us to make that attribution. However, this does give some sense of the ramifications of the different methods of hunting.

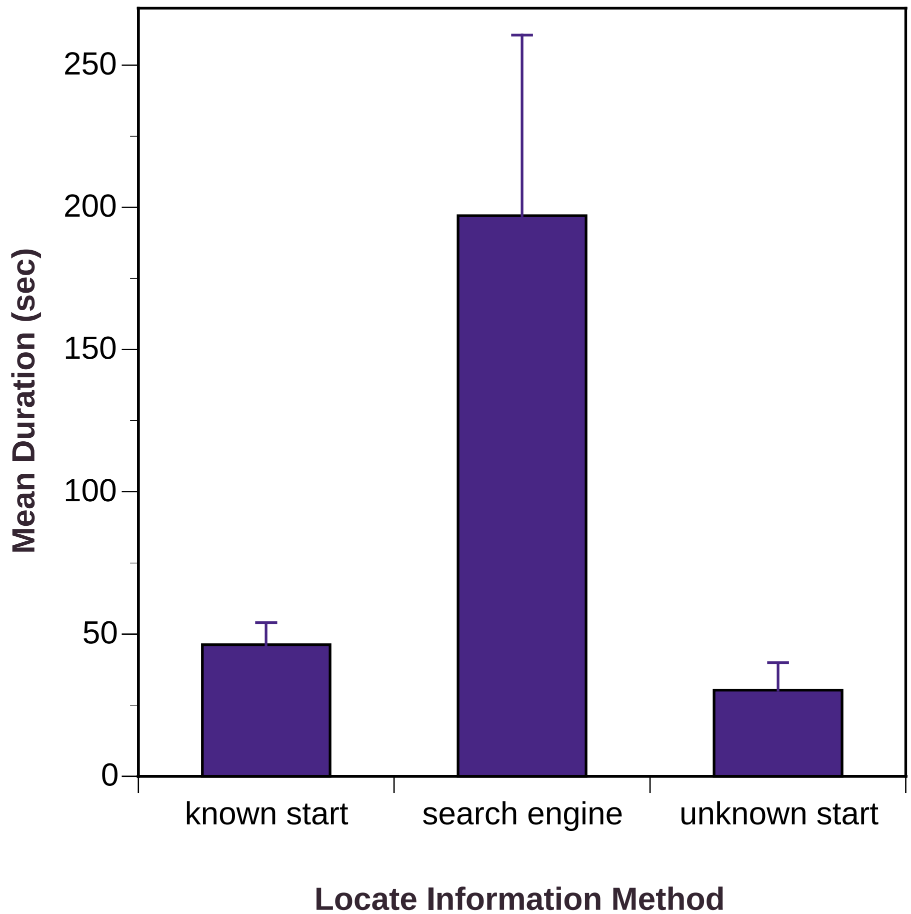

Figure 8. Mean duration for each Locate Information method. Error bars indicate one standard error of the mean.

Locate Information type was associated with method. Table 1 presents the joint frequency of occurrence for types by methods, and this association is reliable (chi-square(6) = 36.55, $p < 0.001$). In particular, hunts targeted at specific pages tended to start at a known locations. This provides strong evidence that users know something about where the pages they frequent lead. This may be, in fact, a driving reason that people include certain pages in their bookmark sets or place certain links on their home page. On the flip side, no searches that were aimed at specific pages began in unknown locations. Furthermore, people seem not to use search engines to browse for things of interest, but instead use them for more targeted searches. Note that a two-way analysis of duration based on type and method is impossible since several cells are empty.

Table 1: Type by method contingency

	Known start	Search engine	Unknown start	Total
Interesting	18	0	5	23
Related	16	7	27	50
Specific	28	4	0	32
Tagged	11	5	9	25
Totals	73	16	41	130

A vast majority of the hunts were successful, 113 against 17 failures (no Locate Informations were coded as “changed”). Given the small percentage of failures, conclusions about success rates of particular types of hunts should be seen as preliminary. However, both method and type were associated with success. Table 2 presents the joint frequency for type and success and Table 3 presents the joint frequency for method and success. Both of these associations are reliable (for type, chi-square(3) = 9.49, p = 0.02; for method chi-square(2) = 7.10, p = 0.03).

Table 2: Type by outcome contingency

	Failure	Success	Total
Interesting	7	16	23
Related	5	45	50
Specific	1	31	32
Tagged	4	21	25
Totals	17	113	130

Table 3: Method by outcome contingency

	Failure	Success	Total
known start	10	63	73
Search engine	5	11	16
Unknown start	2	39	41
Totals	17	113	130

The association between type and success is primarily driven by hunts for pages that are “interesting”--a larger than number of these searches fail than an expected value calculation would predict if these features were independent. The secondary source is the high success rate for hunts for specific pages. The association between method and success is also driven primarily by two contingencies: search engine hunts fail more often than predicted and hunts starting from unknown locations fail less often than expected. Reasons for the latter are unclear, but the former result does not speak well for search engines. These associations are important because there was a large difference in duration between successful hunts and unsuccessful hunts. Figure 9 shows the average hunt time for successful and unsuccessful hunts. Clearly, failed hunts took much longer--the difference is nearly a factor of three, and this difference is reliable, $F(1, 128) = 8.55$, $p <$

0.01. So, while failed hunts were not particularly common, the time cost for a failed hunt was substantial.

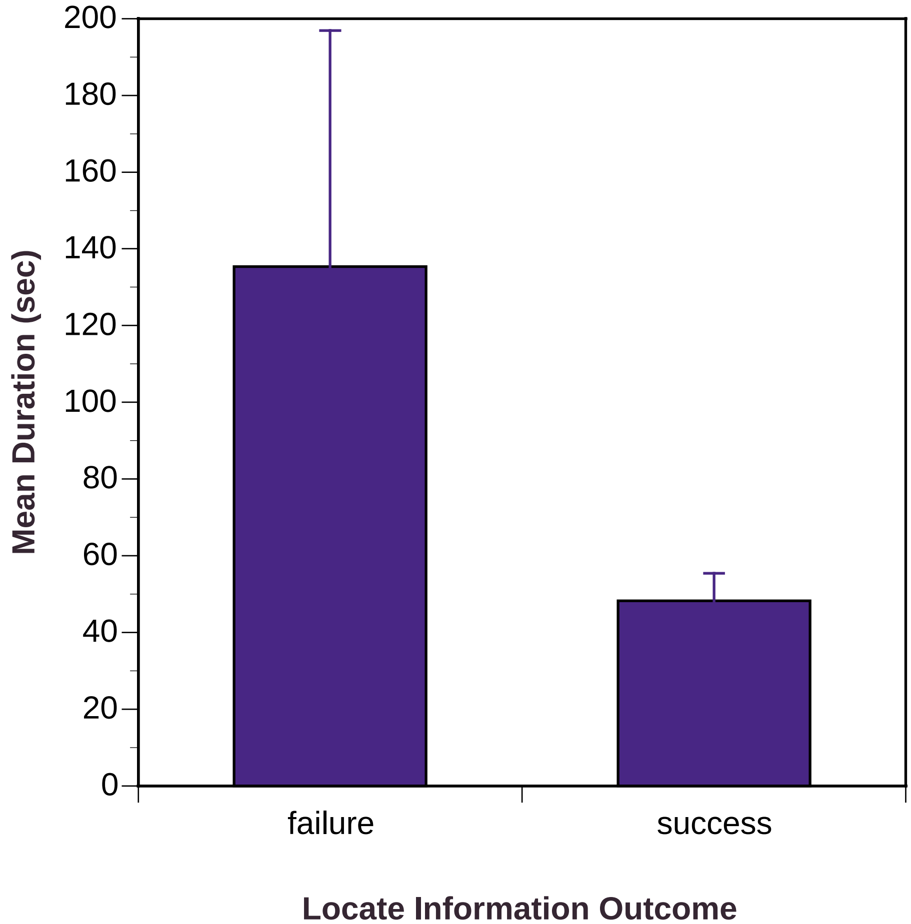

Figure 9. Mean duration for each Locate Information outcome. Error bars indicate one standard error of the mean.

4.3 Find On Page

Find On Page is a more local search than Locate Information; this is the search for something on a single Web page. We observed 588 Find On Page tasks, where were classified by type (string, related, interesting, image, or tagged) and outcome (success, failed, changed). Nine of these could not be categorized by type and so these were excluded. For the 579 remaining Find On Page searches, the average duration was 20.41

seconds, ranging from less than a second to just over six minutes with a standard deviation of 33.95 seconds.

Searches for something on a page related to the target concept were by far the most common. Frequencies for each type of Find On Page are shown in Figure 10. Related searches were more than three times more common than all other types of searches combined. However, there was no reliable difference in duration for the different type of searches, $F(1, 574) = 1.36$, $p = 0.25$. Mean durations for each type of search are presented in Figure 11.

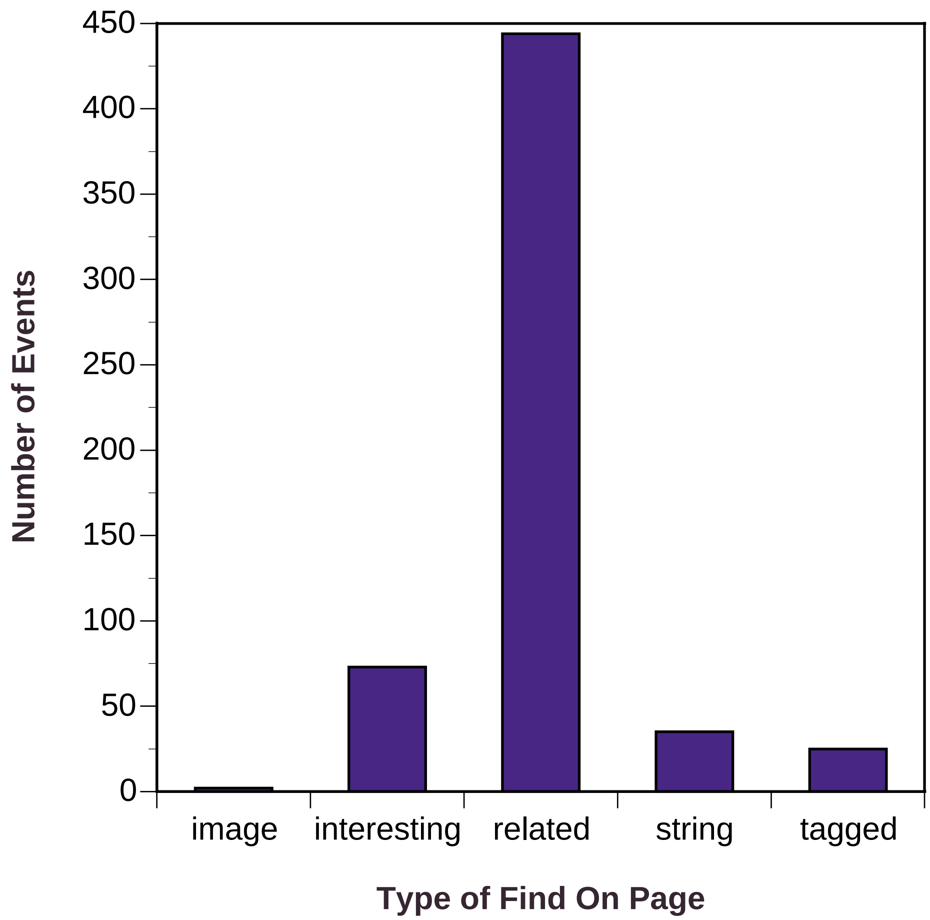

Figure 10. Number of events for each type of Find On Page task.

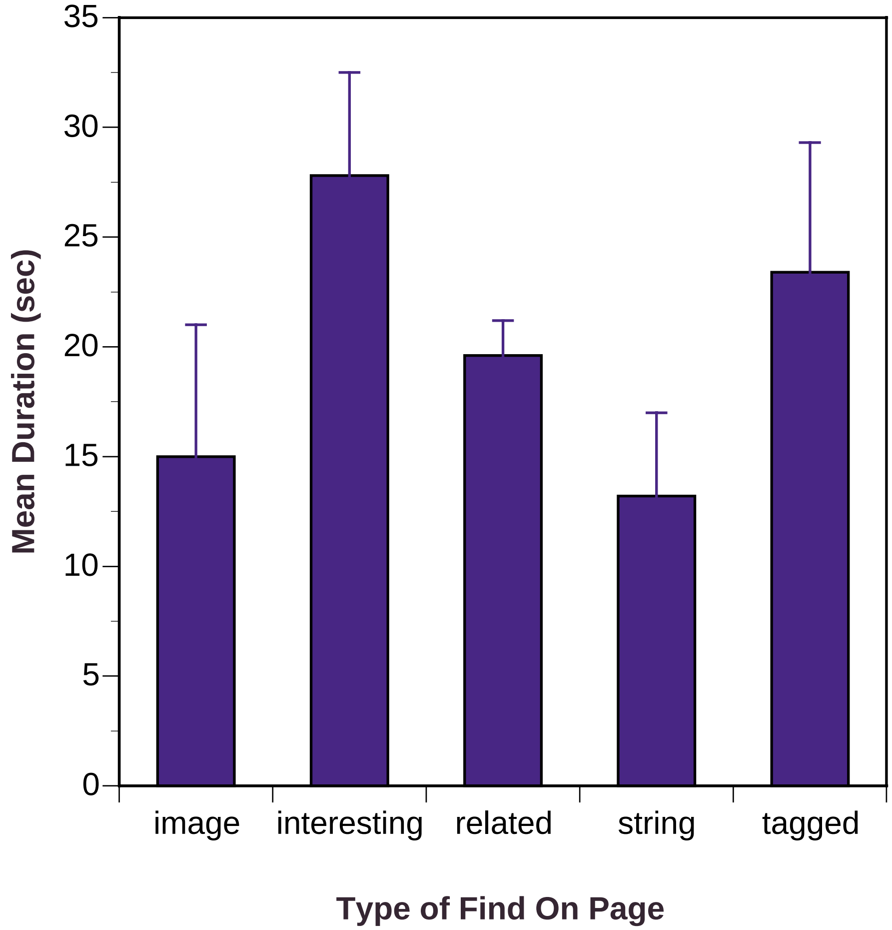

Figure 11. Mean duration for each type of Find On Page task. Error bars indicate one standard error of the mean.

Users were successful more often than not, finding something meeting their criterion on 448 of the 579 searches, a 77% success rate (none of the searches were classified as “changed”). Type of search and outcome appear to be independent, chi-square(4) = 6.50, p = 0.17. Outcome did, however, affect search time. In this case successful searches took longer than failed searches, F(1, 577) = 5.18, p = 0.02). Means for the different outcomes are presented in Figure 12. Reasons for this are unclear.

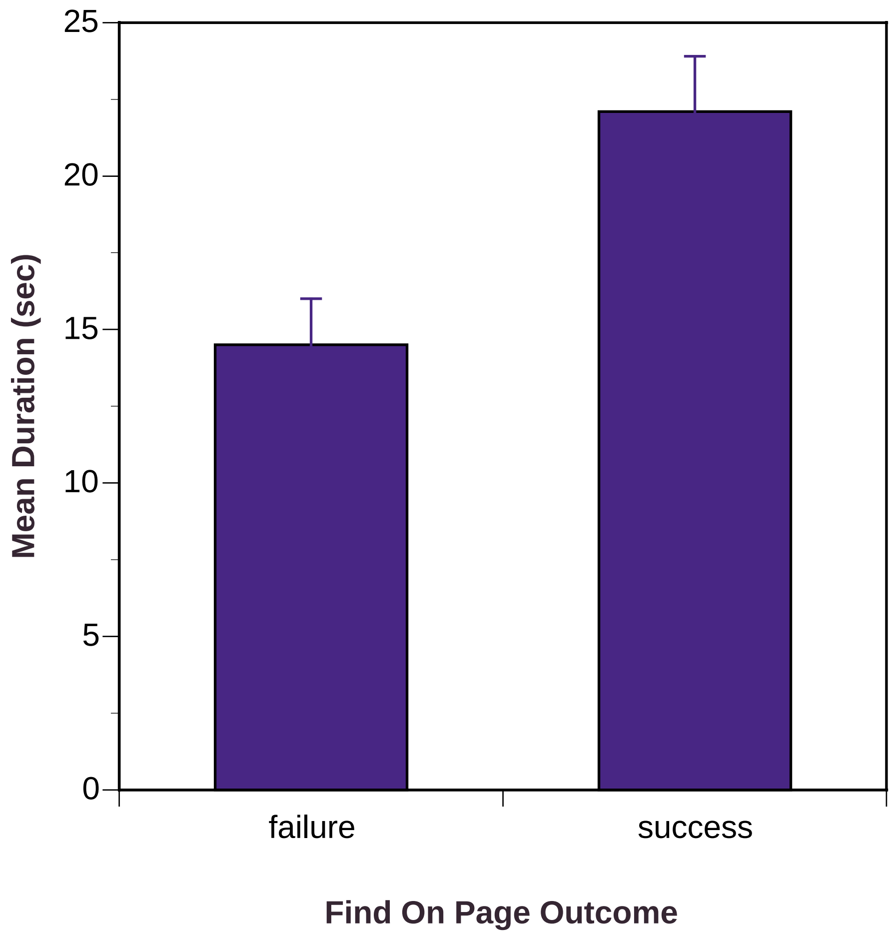

Figure 12. Mean duration for each Find On page outcome. Error bars indicate one standard error of the mean.

4.4 Navigate

Navigating the Web entails going from one page to another. We found 662 Navigate events, of which ten were unclassifiable. For the remaining 652 episodes, the average duration was 8.38 seconds with a minimum of less than a second and a maximum of 3.22 minutes with a standard deviation of 13.43 seconds.

There are a myriad of types of Navigates, only some of which were actually observed. However, two types clearly dominated all others: using the “back” button and navigating by hyperlink. The “other” category here includes bookmarks, external launches, the "home" button, typing in URLs, and a few other isolated uses of things like the history menus. Figure 13 shows the frequency for each class of navigation.

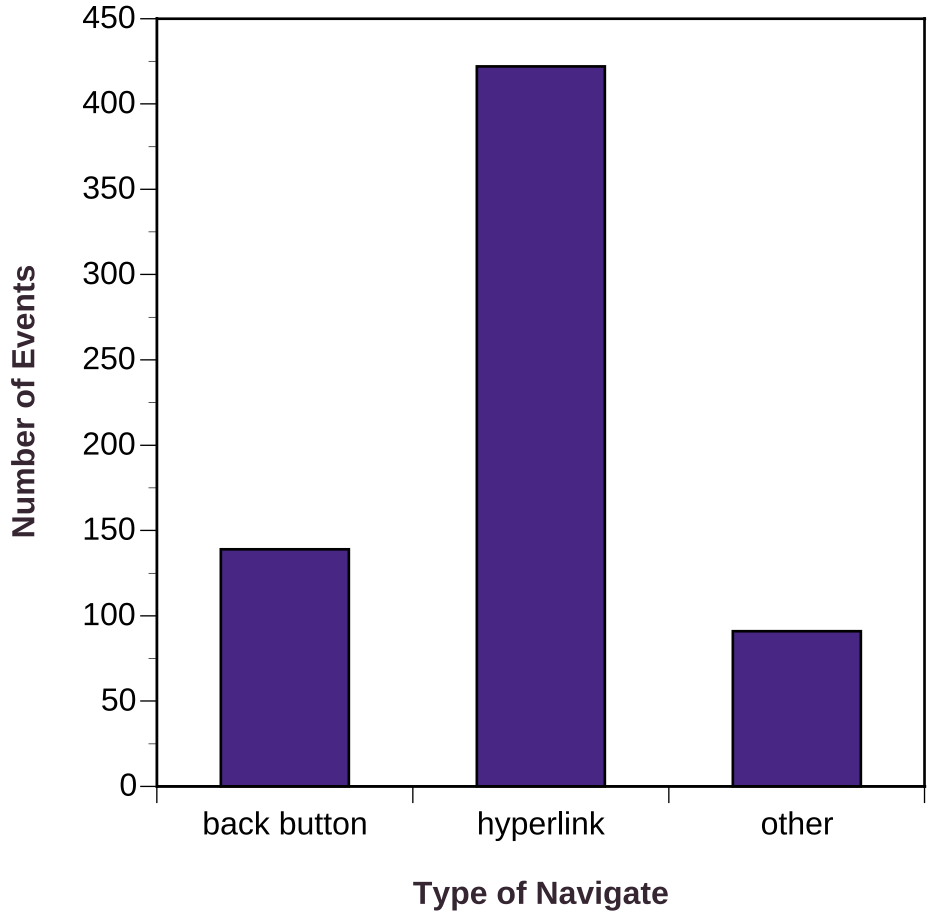

Figure 13. Number of events for each type of Navigate task.

The type of Navigate had a clear impact on the duration, as shown in Figure 14. This difference is statistically reliable, $F(2, 649) = 46.18$, $p < 0.001$. The "other" category being the longest is hardly surprising in that many of these involve typing in the URL. What is distressing is the absolute magnitude of all of these. Navigate tasks include not only the time taken to type or point to the button, but also include time spent waiting for a page to load. That fact that the "back" button takes an average of four seconds indicates that this delay is not entirely a result of network time, because the backward page should be cached. Thus, reading from the cache and rendering still takes seconds. Navigating a hyperlink took, on average, 6.7 seconds--only 2.7 seconds longer than the use of the "back" button.

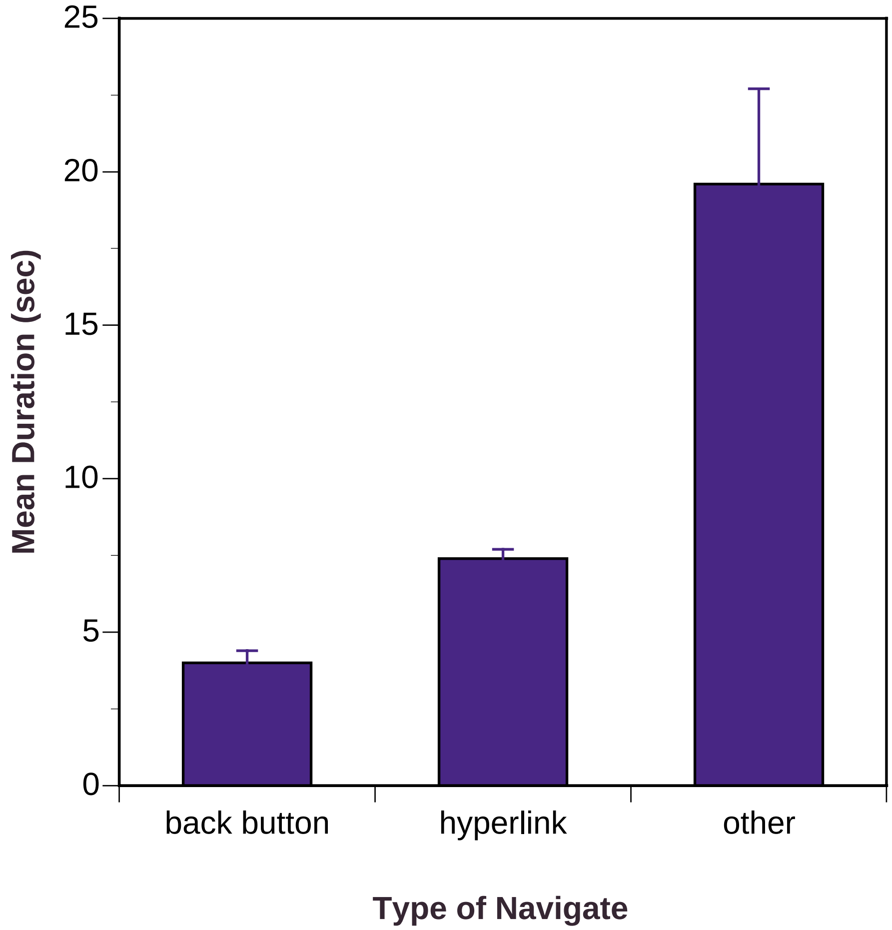

Figure 14. Mean duration for each type of Navigate task. Error bars indicate one standard error of the mean.

In fact, of the 5 hours and 55 minutes spent browsing by our users, 1 hour and 8 minutes was spent simply waiting for pages to load--more than one-sixth of the users' total time was wasted waiting. This is a conservative estimate at best, since these users were all using reasonably current hardware and all but one of the users were on the Carnegie Mellon campus ethernet. Carnegie Mellon is served by a T3 connection, so network response is usually very good. We would expect that most users would not see network response time this good, especially those with modem connections.

4.5 ManageWindow

Users also spent considerable time managing windows, in particular, scrolling. We found 881 instances of window management, all of which could be classified. However, these were nearly all the same kind: scrolling. Scrolling accounted for 728 of the 881 window management episodes. The average time for each scrolling episode was 3.6 seconds, for a total time of 48.8 minutes out of a total of 4 hours and 47 minutes of non-waiting browsing time. That is, users spent about 17% of their active time simply scrolling the page. This is in contrast to previous publications which claim that people are reluctant to scroll (Nielsen, 1996). We hypothesize that users are more motivated to scroll when they're doing tasks that arise in their normal work day as opposed to tasks given to them by researchers.

4.6 Other tasks

While we had several other categories in our Taskonomy, tasks of those types were infrequent and took little total time. Table 4 shows the other tasks and their total and average duration. Clearly, while sometimes critical as subgoals, these tasks were incidental in terms of both number and duration.

Table 4. Other task categories. Total duration is in minutes, average duration in seconds.

	Number of Tasks	Total duration	Average duration
Configure	4.0	4.0	59.0
React	42.0	5.9	8.0
Provide Information	31.0	10.3	20.0

5 Implications for design and evaluation

These naturalistic data have implications for design in many aspects of the WWW. We begin to explore a few and suggest future work that would help contribute even more design guidance.

Overall, the clearest point that these data make is that WWW browsing is a complex mixture of a variety of behaviors, and any attempt to improve the interface to the Web needs to be sensitive to this variety. In particular, our data seem to contradict previously published data about scrolling and reading behavior. However, the apparent contradiction may stem from the fact that the previous results were obtained from studies where the researchers gave the participants tasks, whereas our results came from the participants doing their own tasks. If such different results are caused by these different empirical paradigms, people evaluating their web systems should be extremely cautious to test their designs in realistic settings.

5.1 Implications for WWW Browser Design

What these data suggest is that time spent worrying about things like button layouts and history menus may not have much impact on normal Web browsing. Users do not spend a great deal of time interacting with the GUI widgets of their browsers relative to the amount of time they spend engaged in things like reading, visual search, and waiting. On the other hand, this may well be because the functionality or interface provided to users to support their tasks are poor. It is not clear whether users would spend more time interacting with GUI widgets if they were better designed. For example, we observed little use of the history system. This may be because the history system is poorly designed, as suggested by Tauscher and Greenberg (1980). However, it might also be the case that users would make little use of history systems no matter how implemented.

An obvious case where widget design could make a difference is scrolling. Users spend a great deal of time scrolling (approximately 40 minutes in our 5-hour sample was spent scrolling), and advances which reduce the latency of scroll operations (such as wheeled mice like the Microsoft IntelliMouse) have the potential to save users considerable time. Whether such devices actually do save users time is still an open question, but the potential is clearly there.

Because users spend so much time waiting, improving the performance of the caching and rendering algorithms in browsers should clearly be a high priority as it could potentially save users considerable time. Improving system performance to reduce waiting time is hardly a new suggestion in HCI; however, this appears particularly salient in the case of the Web. Even pages that clearly should have been cached (e.g. those loaded by the “back” button) took an average of approximately five seconds to be fully loaded and rendered.

5.2 Implications for Page Design

Users are willing to scroll through and read long passages, despite claims to the contrary (e.g., Nielsen, 1997) based on “classic directed tasks.” In undirected situations, if users find essays or articles that are of interest to them, they do read them. This suggests that long, textual Web pages are not necessarily a bad idea but should be designed for readability. On the other hand, users do spend a great deal of time searching pages for items related to a target concept, and there may be tradeoffs between readability and “scanability” of a page. These data suggest that the tradeoffs should be carefully evaluated. For some pages it may indeed be worthwhile to sacrifice readability for searchability—but for other pages this may only distract and annoy users.

Some of the initial decisions made in designing browsers defaults were excellent. For example, most Web browsers underline and color links, which can be a tremendous aid to the visual search process—visual search for a target that can be discriminated on the basis of color alone are typically very rapid (Triesman & Gelade, 1980). However, HTML now allows designers to override this and make link colors different than the defaults. Most page design guidelines advise against this, and our data is in agreement with this guideline—anything that slows visual search is likely to cost users time.

5.3 Implications for search engine design

There were three clear results pertaining to search engines in these data. First, our users did not use them when searching for "interesting" information, only when they were much more goal-directed. Second, hunts using a search engine failed more than expected. Third these failed searches took a very long time.

The first result implies that the current search engine interfaces they not be supporting the range of tasks they hope to support. That is, some search engine interface designs suggests they are addressing the more leisurely user, with the prevalence of advertisement, news and sports categories, and other features beyond directed search. However, as with the history mechanism discussed above, these data did not reveal whether the observed use is caused by current flaws in the search engine design or whether it is a fundamental of Web user behavior.

The second results may have many causes: faulty search algorithms, a misunderstanding of how users form queries, some flaw in the way query results are displayed. The actual content of the searches and their results would have to be performed to understand the implications for design.

The third result is probably because a successful search is self terminating (the users stop as soon as they find what they want). However an unsuccessful search is difficult to terminate. As long as the search engine returns additional possibilities the user may continue to explore them.

5.4 Future work

Although the summary data presented here give a high-level view of what people are doing when they browse the Web, the verbal protocols hold a wealth of detail. The current analyses are clearly limited. Future analyses can and should include analyses at higher levels of abstraction (e.g. strategies and patterns of behaviors), and analysis of the contents of the tasks in which users engage rather than just the behaviors. Integration with click studies, which can provide more detailed information about the exact contents of the Web pages being browsed (e.g. “what percentage of the links on a given page are visited?”), is also likely to provide further insight into browsing behavior.

We do not yet know what the implications for design are of the loose hierarchy founded these data. More detailed transition and frequency analysis may reveal prevalent and stable patterns which might be used to anticipate and support common user actions.

Furthermore, our sample of users and environments is clearly limited. A wider sampling of users, browsers, and network environments would not only improve the generality of the results, but allow for more careful consideration of individual differences. We expect that more detailed analysis of naturalistic studies such as this one will provide considerable design guidance.

6 Acknowledgements

We thank Joe Chellman for his help digitizing and analyzing videotapes. This research was sponsored by the National Science Foundation (NSF), Award #IRI-9457628 and by and by the National Institute for Mental Health (NIMH), fellowship #2732-MH19102. It was also supported by generous contributions from the Xerox corporation. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Xerox, the NSF, the NIMH, the U.S. Government, or any other organization.

7 References

Byrne, M. D., John, B. E., Wehrle, N. S., & Crow, D. C. The tangled web we wove: A taskonomy of WWW use. *Proceedings of CHI, 1999* (Pittsburgh, PA, May 15-20, 1999) ACM, New York. pp. 544-551.

Catledge, L. D., & Pitkow, J. E. (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/, Darmstadt, Germany.

Cypher, A. (1986) The structure of users' activities. In Norman, D.A. and Draper, S.W., (eds.) User Centered System Design, pp. 243-263.

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, Held at NIST, Gaithersburg, Maryland, USA, June 3, 1999. http://www.itl.nist.gov/iaui/vvrg/hfweb/proceedings/greenberg/index.html

Lynch, E., Palmiter, S., & Tilt , T. (1999) The Max Model: A Standard Web Site User Model. Proceedings of the 5th Annual Human Factors and the Web Conference, Held at NIST, Gaithersburg, Maryland, USA, June 3, 1999. http://www.itl.nist.gov/iaui/vvrg/hfweb/proceedings/lynch/index.html

Morkes, J., & Nielsen, J. (1997). Concise, SCANNABLE, and Objective: How to Write for the Web. http://www.useit.com/papers/webwriting/writing.html

Navarro-Prieto, R., Scaife, M., & Rogers, Y, (1999) Cognitive Strategies in Web Searching. Proceedings of the 5th Annual Human Factors and the Web Conference, Held at NIST, Gaithersburg, Maryland, USA, June 3, 1999. http://www.itl.nist.gov/iaui/vvrg/hfweb/proceedings/navarro-prieto/index.html

Nielsen, J. (1994) Report From a 1994 Web Usability Study. http://www.useit.com/papers/1994_web_usability_report.html

Nielsen, J. (1996). Top Ten Mistakes in Web Design. http://www.sun.com/columns/alertbox/9605.html

Nielsen, J. (1997). How Users Read on the Web. http://www.useit.com/alertbox/9710a.html

Peck, V. A. & John, B. E. (1992) Browser-Soar: A cognitive model of a highly interactive task. In Human Factors in Computing Systems: Proceedings of CHI 92 (pp. 165-172). New York: ACM Press.

Sanderson, P., Scott, J. Johnson, T., Mainzer, J., Watanabe, L, & James, J. (1994). MacSHAPA and the enterprise of exploratory sequential data analysis (ESDA). International Journal of Human-Computer Studies, 41,633-681.

Safonov, A., Konstan, J., & Carlis, J. (1999) Towards Web Macros: a Model and a Prototype System for Automating Common Tasks on the Web. Proceedings of the 5th Annual Human Factors and the Web Conference, Held at NIST, Gaithersburg, Maryland, USA, June 3, 1999. http://www.itl.nist.gov/iaui/vvrg/hfweb/proceedings/greenberg/index.html

Tauscher, L., & Greenberg, S. (1997). Revisitation patterns in World Wide Web navigation. In Human Factors in Computing Systems: Proceedings of CHI 97 (pp. 399–406). New York: ACM Press.

Triesman, A., & Gelade, G. (1980). A feature-integration theory of attention. Cognitive Psychology, 12, 97–136.