

Introduction

We performed a formative evaluation with users Dave and Jaska. We gave them the tasks of: locating a previously visited page they had viewed in the last week but had found off a daily updating site; locating a previously visited page that they found originally using Google; and locating a previously visited page last semester.

Dave

Task: Locating a page you viewed in the last week

Double-clicking vs. single-clicking/unclear mode change

When stepping through the task of accessing a page in the history, Dave was not happy with double clicking history items; he wanted to single click. The motivation behind this was that he thought the history page was actually a web page. He wanted to use the back to return to the page he left browser mode from. Since our mode change looks pretty much the same and takes over the browser web page area, this mistake is easily made. Screen space is a premium and keeping the mode change is still desirable. We can make the mode change more obvious by adding a transition, such as a drop down sheet, or possibly making the history background semi-transparent over the browser page.

Green highlight has inverse actions applied to it in different modes:

When stepping through the task after Dave realized that the highlighted green \mathbb{H} button meant that he should click it, Dave instantly thought that the highlighted green page thumbnails in history mode should also be clicked. Since the green page thumbnails are indicating instances of the page the user is currently on in the history this is not the correct action. We tried to get double usage out of highlighting the \mathbb{H} we wanted an indicator to link the current page in browser mode with matching pages in history mode and wanted to try and coax people into using the history. We didn't realize the problem that presented. Dave was also confused for a bit in that he thought the history mode could only be used when it was green. It appears that the highlighting the \mathbb{H} button for suggested action may be hurting more than it helps. We could possibly highlight the window border and the thumbnail borders for the association of current page and history page, or we could have the current browser page title or URL highlighted and have the title or url be highlighted in the matching history pages to be more obviously matches to the current location.

Task: locating a page from a previous Google search; get information that has been previously collected when little/nothing is known about the path taken or any sites involved

History returning to it's original state

When walking through this task, Dave wanted to know how to return the history to it's original state after a search term had been entered since his list was now filtered. He thought the back button would work, since his original preconception was that it was a webpage. Since hitting the back button puts you back

into browser mode on the page you were on just prior, it was not the correct course of action. It was hard to illustrate with the paper prototype that the search was filtering the sessions as he typed in them in, he said if he had realized that it wouldn't have been an issue since most Mac OS X applications have that behavior. Adding a \otimes button to the right edge of the search field for clearing the filtering would also help in this regard as it is consistent with most Mac OS X applications, even though it was omitted in our paper prototype.

Task: locating a previously visited page that you last viewed a semester ago; get to a well-defined site that has been previously visited when a path to this site (an exact sequence of steps to get there) is known

Session chronology not always apparent

When discussing the issue of locating pages in history, Dave figured that the sessions were sorted newest to oldest, but it became hard to distinguish when these sessions took place, and he wasn't sure about the ordering when searching and sessions are filtered out. He wanted to be able to scroll to the previous weeks easily or know when a session took place in a search result. We did overlook specific dates for sessions, figuring that after a while, really old pages' specific dates wouldn't be remembered and the search filter would be enough to narrow the number of sessions that dates wouldn't be needed to distinguish. It seems that searching shouldn't have to be necessary, and that we could add a date tool tip while the user scrolls to easily navigate by date to get to a certain time frame (like how adobe acrobat displays page numbers when scrolling). Also having the date along the top edge of each session would give users an affordance to realize that the sessions are filtered but still chronological and not reordered by relevance.

Jaska

Task: Locating a page you viewed in the last week; get information that has been previously collected when little/nothing is known about the path taken or any sites involved but something about content

We got to see Jaska's curious approach to the new interface as he performed this first task. After we asked Jaska to pull up the 2-week old Slashdot article, he by default said he'd type the URL for Slashdot in, but stopped short and

“Wait! Is that \mathbb{H} history?”

This being his first experience with our interface, it's important to capture whether or not it maps to what he expects. Jaska initially guessed that the \mathbb{H} button did something with history, perhaps due to its carved stone look, but he does already have some insight to the project. Our interface isn't a standard browser feature, so it's a challenge to inform a user that the browser isn't completely different, just enhanced. He expects a mouseover to describe what the button does, a sound idea.

“Do they have dates anywhere?”

(looking at full history to maybe try and pick out) Our paper prototype didn't include such detail, but dates for each session was something Jaska was intuitively looking for, and was noticed missing even at this prototype stage. Moving toward more real prototype, note to make sure dates are obvious, not hidden in a special feature or mouseover. An important, if not the strongest affordance for Jaska was that a recognized pairwise match somewhere in path leads to guaranteed match for goal; this applies to full and trimmed history.

Search History input clues

(was too many results to find in big history, trim results) One of the potential issues with text input in an interface is that you don't constrain input. For our interface, all input is valid, but in this case Jaska thought only certain things would work, “what goes there. can i type a URL, or a name?” If a user thinks he/she can only type in a web address and it works fair enough, they might never realize they can put other key words. This isn't Netscape 6.0 where bookmarks are old-hat, pretending this is bookmarks in Mosaic, perhaps the default term for the search box default to “Enter some terms to search history” instead of 'Search History'. 'Search History' may be more appropriate when all browsers have the function and users know more of what to expect.

Expected output following search History; “Does it pick an order to sessions?”

(find session in trimmed results) Jaska wasn't sure exactly what form of result output to expect from searching for 'slashdot'—what would his new information change? After searching, he said he wouldn't look for the date any more, but care more about a result's relevance to his search term. He took the time to think of a search term, so our interface should maximize the return. Bridge his expectation to reality by communicating a session's relevance to search (order by relevance, combination of date and relevance in order, pop-up to choose result order?).

How does it know when to turn green

Should H really turn green every time he goes to slashdot? The question is, is the green glowing all the time anyway? He goes to a lot of the same sites each day, if it's always green, what's the point. If we choose to grind at making this feature work, which is doable, it will need some very smart code to know when to turn green. Every user can have different browsing habits.

Task: locating a previously visited page that you last viewed a semester ago; get to a well-defined site that has been previously visited when a path to this site (an exact sequence of steps to get there) is known

Status Quo May Still Be Preferred

We asked Jaska to get the engineering Gradcheck site as we have done several times already. As we discovered and reported earlier, Jaska has a set method for accomplishing this task which involves entering a URL, then clicking 3 specific links. Jaska began this task as we expected: he entered `www.ece.uiuc.edu` into the address field and clicked enter. He was intrigued when the \mathbb{H} glowed green, so he decided to click it and quickly found (and clicked on) the proper page.

Having completed the task, Jaska explained that he liked the idea that we were working on and commented that he might even use this history feature, but not for tasks of this manner. Jaska feels absolutely certain that his current sequence of links will get him to the Gradcheck, but has no idea if the “mysterious green button” will. Jaska did not feel that saving “a click or two” would be worth the uncertainty that comes with trying history. Further, using the feature would involve a comparatively difficult visual search/recall task rather than his memorized pattern of clicking. We either need to reduce the scope of our project to not include this task or we need to make sure it develops enough trust with the user. A major fear is putting out another clippy, so have it only be loud when it should be; the procedure simple enough that saving “a click or two” seems worth-while.

Task: locating a page from a previous Google search; get information that has been previously collected when little/nothing is known about the path taken or any sites involved

Must Reduce the Gulf of Execution

We asked Jaska to find pricing information about creating custom printed T-shirts (a task that he had already done twice). He said that he would definitely click on the \mathbb{H} button because the subject was “so random” for him, not something done often. He felt that he would never remember the Google search term he had used. Clicking on the \mathbb{H} brought Jaska to the entire history (a list of all session histories in chronological order) which he began to scroll through. Realizing that scrolling would take a very long time, Jaska decided against using our history feature altogether and began the task again by going to Google.

The problem was evident just by watching Jaska do the task. It was not fully evident to him how to apply the small amount of knowledge that he had about his target. Instead of trying to figure out how to use the information in his head to guide the system, he decided to use a guaranteed method—a fresh Google search. Again, we will need to gain the user’s confidence and to make sure that the gulf of execution is as small as possible. The user ideally should be able to do anything he wants with any information he may have, especially executable by searching and scanning the history with what he knows and can pick out. It presently seems like only experience with the new system will lead to the expertise necessary for it to be helpful in all tasks. The more a user gets to know how our system behaves, the better he can predict, and the more effective the system will be. If we know the biggest improvement will be by increasing the speed that trust is earned, we know to design for learnability.

So now at Google, Jaska came up with a search term (ended up being different than before) hoping to get similar results. Here, we diverged from our actual prototype plans to see how maybe a different criteria for glowing \mathbb{H} would fly with Jaska. For Jaska, we made it glow because it reached a threshold of URL and keyword matching (not just the same URL).

Trust

“There’s no way the browser is this smart. How does it know Google results pages? Aren’t those on the fly and always different?” Though he had this confusion, he said he’d actually give History a shot again since the green would be visually pleasing and he could scroll the results easy enough anyway (yay for mouse scroll wheel). Here, it is evident that Jaska expected the \mathbb{H} to only glow green when a URL was matched. A user might quickly realize the \mathbb{H} glows green when our code is satisfied that a page’s frequent words match up with enough frequent words occurring in history, but it is a risky thing to do. No matter what we implement, the mouseover tooltip of the \mathbb{H} button when in a glowing state should be more descriptive than normal and explain *behavior*.

Improving the Interface

Our interface begs to be explored. It has a behavior that depends a lot on the users habits, and the way for that behavior to become predictable for each individual is for them to explore it. It will help to keep in mind to make it engaging, as to pick up and keep users’ curiosity.

- Even if history mode shows up in a modal sheet or semi-transparent overlay, it makes sense that widgets other than \mathbb{H} and the search field should manipulate the history mode as well rather than take the user back to browser mode. The history representation is a record of what goes on in browser mode, so then mirroring action in both browser and history contexts when using the back/forward buttons (such as having the location highlight move on the current window session in the history while the background browser view changes as well) seems like a reasonable idea, and then leaving mode changes to the toggle \mathbb{H} button and double clicking thumbnails.
- Jaska thinks linking history to bookmarks is really important. We weren’t sure if doing bookmarks would be crossing over into adding browser features rather than working on our history interface. Having an easy way to bookmark history items, such as selecting thumbnails in history mode and choosing or clicking a bookmark adding button would suffice for history interface. In addition, we thought being able to bookmark sessions or lassoing portions of sessions and having them represented as a path in bookmarks, similar to our history representation, would be beneficial to the user, but could be outside the bounds of the interface we are trying to improve.
- Most of the usability issues discovered in each of the individual tasks will help all history tasks, as there is a great deal of overlap.