Joshua Benjamin, Blair Flicker, Nadia Karlinsky, Jay Tuley

User Evaluation Results

During the whole scope of interaction in TrailBlazer, a user switches within several modes of thought and task. Browsing is a highly variable task mode; different websites have different modes of interaction. However, we can generalize the majority of browsing as either information-seeking or casual browsing.

TrailBlazer increases the functionality of a traditional browser, allowing the user to see the relationship between sites they have visited. When a user toggles into history mode, things look different; the eyes must adjust. Not only does it look different, the psychology is often different between the two modes. Compared to existing browsers, TrailBlazer has much higher cognitive difficulty and intensity.

Even more of a seeking task than web browsing, the user will often know the exact desired target. Signal detection task among noise; there is a lot of noise, so there are several methods to reduce it. The user will then drill down into the session and area of the history where he/she knows the target lies. At this point, it is still a search task, but different: usage of tooltips and zoom for larger thumbnails increase exponentially.

Our design includes an attempt to establish a new meaning for color, which has been very difficult. The notion of color signifying important information can be difficult and dangerous. For this reason we decided to limit the number of meaningful colors in our application to only one – green was to imply current site. Falling victim to the egocentric intuition fallacy, we felt it was only necessary to reinforce the color significance because we all understood what it was supposed to mean. We learned that clearly establishing a new association is significantly more difficult than reinforcing a known one. We will only use color in the future if it seems very necessary.

Usability Issues

1. Visually cluttered
   a). Critical – There was visually too much noise, too much for you to easily scan to what you want.
   b). Implementation Very Hard- Writing a visualization from scratch and enhancing it is very difficult. Space things out and making the branches collapsible would help remove the effect of the noise. Also adding better algorithms for combining nodes for things, like search result pages or

duplicate pages in a row would help reduce clutter.

c). We won't have time to implement in the scope of the course, but this is one of the things that we would implement with time, as the visualization is one of the most important parts of the project.

2. Row delineation is poor

a). Critical – this made it difficult for the user to realize what web paths corresponded together and was due to too much transparency. LCD screens also have lower contrast, which makes it even harder to distinguish.

b). Plausible reasons: the transparency makes the standard OS X rows too subtle.

c). Easy – Removing transparency, it was already difficult for the users to notice that they could see their current page and it confused and distracted more than it helped. The non-transparent color matches Apple's UI guidelines, however it still seemed too light; darkening the color is a one-line change.

d). This is something we can do near-term.

3. Italic Date Grabbed Attention

a). Low criticality- It doesn't hurt the interface that much and is not something that needs more attention than other things.

b). Easy- Removing italics is easy enough, to help set it apart from the thumbnail we can space it a little farther away.

c). This is listed because it goes along with visual clutter. Increasing the size of each cell may be doable near-term, but could become time-consuming. Give it a shot.

4. Can't visually investigate pages easily

a). Critical- as this was a task we wanted to support.

b). Solution implementation is tricky, these solutions, themselves, bring a host of possible new interface problems with them.

c). Possible choices involve doing MacOS X dock-style magnification or adding image to the tool tip.

d). A small step to fixing some of the issue might be to have a minimum icon size and thus allow horizontal scrolling; horizontal scrolling would be an easy coding fix. However, this means that the whole session, specifically the tail end, would not be viewable without first scrolling, not a behavior we wish to introduce.

e). Implementation plans: We will start by trying to make a keyboard modifier bring up magnification on mouseover. Adding images to tool tips may not be possible using standard APIs.

5. Tool tip obscures thumbnails

a). Medium criticality- Across our users, there was split whether this was an issue or not.

b). Hard Fix- The nature of tool tips makes it difficult not to obscure any thumbnails.

c). Possible design changes: Having a set location on the screen away would solve this issue, however there would be less association and more eye-jumping around from target to target across the screen. Having a preference to choose between tool tip and on location for rollover information.

d). User evaluations revealed some specific interaction details that fit best in user preferences. This is one of them.

6. Summaries were too long for quick scanning

a). Medium Issue, again it was split between those who thought it was too much information and some who really liked the amount of information.

b). Medium difficulty on implementation, add a preference to account for differences in browsing styles, pretty easy as the summaries are generated by a summary API that takes the text and the number of sentences to output too.

c). Probably not added to the list as user preference varies.

7. Mouse precision hard on small thumbnails (Fitts's law)

a). Plausible reasons: Small thumbnails are a result of not having a minimum thumbnail size. Arrow key movement is reasonable to expect

b). Possible design changes: make thumbs a minimum size, add arrow key movement

c). Severity of problem: medium

d). Implementation difficulty: medium

e). Implementation plans: We wanted to add this before, we didn't have time, it won't require rewriting, but there's still a bit to implement for arrow key movement behavior to work correctly.

8. Pulling up summary is inaccessible without mouse

a). Low – tooltips don’t have other input support, but if we did that’s a draw back of using the tool tip API.

b). Solution is Medium difficulty – have a keyboard shortcut to toggle tool tips

c). Not an immediate issue, as we don’t have keyboard navigation.

9. Search carrying term over between WWW and History modes; has issues with hits and not hits not always easily distinguishable; and Google pages overwhelm all results.

a). Medium– there needs to be more indication when searching.

b). Hard to do – we need to add more to the GUI such as: display the matching search term above thumbnail for each page that matches term, and give google pages a reduced weight.

c). Requires a lot of code work, this belongs on the list of things we would do if we had the time.

10. User expects thumbnail to be a link (single-click)

a). Low criticality – this wasn’t observed for each user, and is a pretty common problem in most applications.

b). It’s hard to solve this problem, keeping double click behavior seems like our best option, it matches with OS X interface guidelines (being an icon and using selection color) and if a person thinks single click is selection and it jumps to the page takes more time to correct the error than single clicking and then immediately double clicking (as our users who had this issue did.)

c). No low-level solutions added to the list of things to implement, as we are not sure what else we can do at the low-level. High-level, the better enforcement of proper mode via look-and-feel, the easier a user will realize how to properly interact.

11. Green Color minimal impact to program functionality: doesn't obviously signify a present page is in history.

Confusion on representation of green versus size.

Green was too light to see in the icon and scrollbar, (thumbnail was ok).

The scrollbar knob often obscured the scrollbar slot's green tick mark.

a). Critical – Users that were involved in our early prototyping had much better success than our one novice user. We had thought originally that we would have fair success by just enforcing a consistent characteristic (the color highlighting green), and that it would work, even if some aspects were chosen without a history related reason. This was an error.

b). Hard--We could make the make the border of the page green which would correlate nicely to the thumbnail also having a green border or we could use a mall map style fat arrow (you are here) to actually have an way for users to easily understand the indicator, a lot of work would be needed to keep it from cluttering the interface. The scrollbar green highlight, was often obscured by the scroll knob, we can use OS X's compositing API's to fix this, probably would use mini arrow icons too, rather than pure highlighting.

c). This is feature is important to our user interface goals, but it's a lot of work as well, complementing the unfinished visualization It is something we should do if we had time.

12. Mental Model of History as a Webpage, Not a Mode

a). This was by far the most serious concern that arose with Dave. He would enter the History mode, but try to use the back button to return to Web mode (as if History mode was another site). Mark felt that the back button would take him to the History because "it was my last view, it felt like the 'previous page." Goes along with wanting to click on thumbnails once because Dave thought of them as links.

b). Medium criticality

c). Resolved by more strongly enforcing idea of being in a different 'space', implement by modifying behavior to more significantly change toolbar interface when in History mode. May be a moot point, as the History function should altogether eliminate the concept that back and forward buttons are supposed to afford. This begs further debate, but the removal of back/forward buttons certainly classifies TrailBlazer into a new interaction paradigm.

Other Information Discovered During User Testing

Privacy Issue

Dave checked his email while using the browser, and like any site, TrailBlazer took a screenshot and displayed a thumbnail.

The privacy issue came up in our initial conceptual discussions. We decided that at some point, we would add a way to make TrailBlazer stop watching. We did not feel this feature was something to concern ourselves with until interface evaluations for a release version. When temporarily disabled, we imagine some sort of indicator that TrailBlazer isn't watching, so that the user doesn't forget it is off and later be surprised by missing data.

Users Rarely Get More than Two Links Deep in Sessions

Because up until this point we have given Dave relatively specific tasks designed to necessitate clicking many links, we had never noticed Dave's natural browsing behavior: he will enter a URL, visit a small number of links, then enter a new URL. He is very sporadic and even returned to a couple sites by re-entering the URL during our short "free-play" browser orientation. This means that he generated a large number of very short session histories, which make for wasted space in our current history design layout.

One of the major investments in the development of TrailBlazer is the ability to see paths taken, but Dave rarely goes far enough on any path to get lost. Still, our goal is to make finding past pages easier, and TrailBlazer is to support all types of browsing behavior.

New Applications

1. Usually switching to the history and checking on a session(s) could quickly tell you whether a bunch of pages that you're trying to load in parallel have completed loading. Especially useful to modem users.

2. The term "Aggressive Browsing" coined. Would be nice to browse and use the history at the same time: split screen or transparency.

3. Can replace or complement bookmarks and tabbed browsing. Mark explained how he might use TrailBlazer history to open his morning webpages each day.

Lessons Learned:
How to Conduct a Similar Study Better in the Future

User Feedback Sometimes Reflects Personal Preferences, Not Good Design

We have now run our users through several rounds of evaluations. At first, we took the feedback and suggestions we got from users as if they were absolutely correct. We have learned that although user insight can be incredibly helpful, it will almost always be very centric with respect to the individual user. One user may love navigating programs using the keyboard while another may despise this paradigm much preferring mouse-based controls. We have come to understand that it is not our position to turn every user suggestion into code, but rather to take all suggestions into account and find innovative solutions such that all users can easily relate to the enforced interaction. The user doesn't have all the answers, that's our job.

Have a Good Mix of Users: Investment, Personality

Mark's evaluation was invaluable. Since Mark was not involved in the paper prototyping and had not ever provided input for the program, we were able to easily find non-intuitive aspects of the interface. Additionally, he had no emotional investment for ideas, so we could trust his comments to be purely objective. He also provided several thoughtful comments about how he would apply TrailBlazer, which we had never imagined. He also offered his own new ideas for how he would improve the interface. His ideas were different from Dave and Jaska's comments, whom were involved, and one could say tainted, by their early involvement in the project.

In the future, it would be desirable to have an even wider spectrum of investment. Different involvement among three sets: some users involved in early observation, some users also trying out the paper prototype, and a separate set that only sees polished versions of the functional prototype. Some users involved in each step from the beginning, some only in one step, etc. Each individual user is then more likely to provide a rich set of input instead of everyone being very similar.

In addition to a wider spectrum of investment, it would be beneficial to have a wider spectrum of academic/professional backgrounds. In the current study, our users were working on engineering degrees. For example, Mark, a graduate student in Electrical Engineering, readily understood the concept of the session tree, which might not be as intuitive to someone without a mathematical background. Mark used his knowledge of the properties of graphs to aid in his search for a particular page in our artificial history, reasoning that he would search for a "more complicated tree . . . dense . . . link will be closer to the child side, not the root side" given the contrived characteristics for the target page. A user not familiar with the properties of graphs might not necessarily use the same reasoning to locate the target page, and observation of such an interaction would be a valuable experience.

Onsite Evaluations Were Often Interrupted by Bustling Living Space

In order to encourage the most natural interaction possible, we conducted the evaluations in the most natural setting for the users – in their homes. We would like to think that this did achieve our goal of observing normal behavior, but we did learn the drawback to our technique – there are many distractions in the place where one lives. We were unable to control the environment; unfortunately telephone calls, interruptions from roommates, dinner cooking, and having distracting personal items close at hand, all made it hard to keep the user's concentration. In the future, we think it would be very wise to seriously consider having users perform evaluations somewhere where distractions can be avoided.

Have Noticeable Bugs Worked Out

Having bugs in our program tended to elicit the wrong kind of feedback. When bugs were found, user feedback was very likely to report broken or incomplete features and functions. That kind of technical feedback is not at all what we want; we need information about how natural and intuitive our interaction paradigm is for them. Having bugs obfuscated the real usability issues. One of our concerns was that searching through the history could demand a good deal of scrolling. We were not able to get a good feel for how users dealt with this issue because the scrollbar often would not show up.

Users May Ask the Evaluators Questions

Jaska really surprised us by asking many technical questions about the application. We did not discuss what to do in this situation, so pride in the project caused the conversation to slip into the realm of implementation details. This was not only unwise use of our limited evaluation time, but it also turned the session from an evaluation into demonstration at times. Jaska then applauded each feature rather than experiencing the application as a whole and evaluating its usability. In future studies, we would need to discuss how we plan to politely tell users that we cannot explain anything to them and that we are interested in how they are able to deal with the program on their own.

Evaluation was Based on a History that the User Did Not Create

The premise of our project was to create a program to find a site that has been visited before as quickly and easily as possible. Due to time constraints, it was not realistically possible to schedule two sessions with each user – the first to surf the web given minimal constraints and the second to find particular sites again – as well as allowing for time between sessions for memory decay. As an alternative solution, we decided to fabricate an artificial history of visited sites and tell users what they "remember" about the target sites. This task was so contrived that the information may be only marginally indicative of real behavior. We learned that we must have more realistic and less contrived tasks in order to get the most useful information.

Joshua Benjamin, Blair Flicker, Nadia Karlinsky, Jay Tuley

Questionnaire Results

	Number of Users That Responded:				
Question	**Strongly Disagree**	**Disagree**	**Neutral**	**Agree**	**Strongly Agree**
The mode change is easy to use	0	0	1	2	0
Seeing the partially opaque website under the transparent History is useful	1	0	0	2	0
The information provided when viewing the History map is useful in locating sites	0	0	0	3	0
The amount of information is appropriate (not insufficient or excessive)	0	1	1	1	0
Searching the History is intuitive/easy to use	0	0	2	1	0
Green highlighting is intuitive/easy to use	1	0	0	1	1
Blue highlighting is intuitive/easy to use	0	0	1	1	1
Different sized icons are intuitive/easy to use	1	1	1	0	0
Overall, the application is natural and intuitive	0	0	2	1	0

Comments:

Going to History was intuitive; coming back from History was not

Transparent is distracting - it is hard to delineate what is in the History and what is on the page behind it

Good amount of information, however placement could be changed

Border of page should be green highlighted as well

No back, no forward