When Two Methods are Better Than One: Combining user study with cognitive modeling

Andrea Knight
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043 USA
aknight@google.com

Guy Pyrzak
SJSU Foundation
NASA Ames Research Center
Moffett Field, CA 94035 USA
gpyrzak@arc.nasa.gov

Collin Green
SJSU Foundation
NASA Ames Research Center
Moffett Field, CA 94035 USA
cgreen@arc.nasa.gov

CHI 2007, April 28–May 3, 2007, San Jose, California, USA.
ACM 978-1-59593-642-4/07/0004.

Abstract

We discuss the benefits of combining user studies and cognitive modeling in the context of Firefox tabbed browsing. We studied new users' ability to use tabbed browsing without assistance, and then evaluated alternatives for closing browser tabs to improve the new user experience through user tests and cognitive modeling. In general, our experience highlights the advantages of using user studies and modeling together to do user interface evaluation: user studies provided validation of design intuitions and data to support modeling of user behavior; modeling provided a fast and efficient ability to play "what if" with the design change; the combination of qualitative user test data and quantitative modeling results proved to be a far more convincing package of evidence than either result in isolation, given the variety of perspectives in the design and development team.

Keywords

Usability Research, User and Cognitive models, Performance Metrics

ACM Classification Keywords

H5.2 Information interfaces and presentation (e.g., HCI) --- User Interfaces: Evaluation/methodology, Theory and Methods

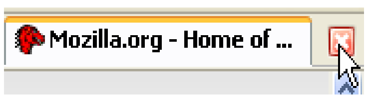


Figure 1. *Corner-X* - Close icon is found in the upper right corner,

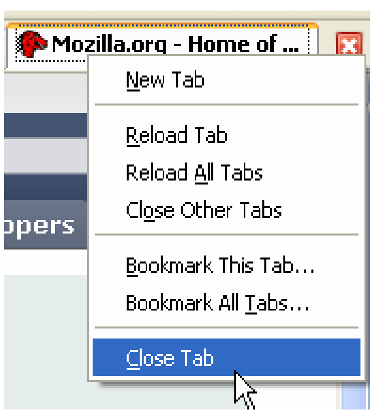


Figure 2. *Context menu* - Close option is found when user right-clicks, or ctrl-clicks on tab

Introduction

Have you ever been approached by a development team for the answer to a hotly-debated user interface (UI) issue? Perhaps everyone involved has well thought out ideas, opinions, and compelling anecdotes, but the team is looking for something definitive to inform their decision? Maybe the team is questioning how a UI can better support new users while still meeting the needs of its current user base. In such cases, a small-scale usability study may not be compelling, or the initial study findings may raise additional questions. Here we describe how we combined small-scale usability studies with cognitive modeling to help Mozilla Firefox developers make an informed decision about tabbed browsing in Firefox [4].

We were approached by developers of Firefox to help understand and improve novice users' experience with tabbed browsing. We began with a user study, and found that we needed to go further to understand how to meet both novice and expert users' needs. Cognitive modeling helped to inform and refine our ideas, and enabled us to communicate usability issues to developers. Finally, a second user study validated the decision supported by our earlier efforts.

Can novices figure out tabs?

Initially, we asked whether novice users can figure out tabbed browsing without assistance. We designed a small usability study (think-aloud) with seven participants to introduce novice users to tabbed browsing in Firefox. Users were paid volunteers from the community and were selected to have little or no experience using a browser that supported tabbed browsing and no experience with tabbed browsing. Users were asked to complete webmail and search tasks using a tab-enabled version of Firefox that forced new pages to open in new tabs.

Seven out of seven novice users introduced to tabs in this way were able to use the tabs successfully. The discovery of tabs in the browser took one of two forms:

- users failed in attempts to return to a previous page via the back button.
- users encountered the "you are closing multiple tabs" warning message.

Not all users immediately identified tabs, understood tabs were part of the browser, or verbalized an accurate understanding of the tab concept. However, by the end of the study all users understood the concept of tabs and could navigate among tabs. Many users reported preferring tabs to their current browsers in post-study interviews. These findings encouraged the team to enable tabbed browsing by default.

"How do I get rid of that tab?"

The user study also revealed that novices encountered difficulty when it came to closing tabs. We found that tab closure was not immediately discoverable for novices. Seven out of seven users initially closed tabs by invoking the context menu. Following probing questions, (e.g., "Is there another way to get rid of the tab?") only two of seven users discovered the *corner-X*. It was hypothesized that users in the study were less likely to discover the *corner-X* because of its distance from the tabs (which were aligned at the left); it was out of their line of sight. The two users who found the *corner-X* hesitated to use it because of their association with the similarly aligned window control. These users expressed concern that the *corner-X* would close all the tabs.

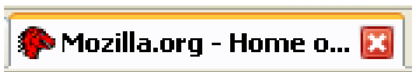


Figure 3. Alternative Tab Closure
Tab-X - Close icon is found on each tab

While users were able to close tabs, there was concern that novices might become frustrated using the context menu to close tabs before discovering the *corner-X*. We decided to evaluate alternative UIs that would improve discoverability of efficient tab closure.

Evaluating Alternative UIs

One candidate UI had embedded close icons in each of the individual tabs (the *tab-X*). The *tab-X* design had two advantages over the current *corner-X* design:

- the *tab-X* design would improve discoverability by bringing the close icon into the line of sight
- the *tab-X* design would strengthen the association of the close icon with the tab (via grouping by proximity and common region, see [5])

However, one concern was that the *tab-X* design might have a negative effect on expert users' experience, adversely affecting the efficiency of their tab closures. We could have conducted another usability study, but instead chose to use cognitive modeling to predict expert performance realistically and equally for comparison of the two designs. Modeling eliminated the need to recruit and run participants, and would give the team more quantitative data to inform their decision.

Predicting Expert Performance

To help the team evaluate the effect of the new UI on expert user performance, we created cognitive models to compare the closure a single tab using three methods:

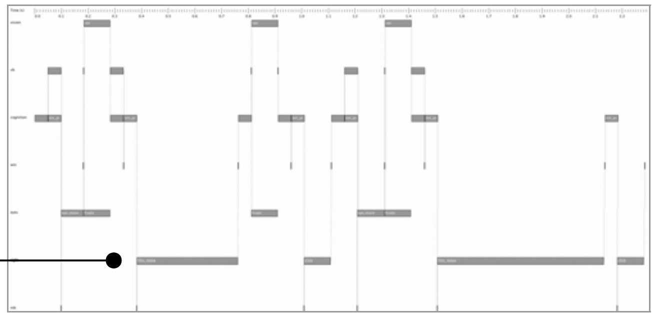
- clicking the close icon at the upper right corner of the browser window (the *corner-X* method)
- clicking the close icons attached to individual tabs (the *tab-X* method)
- clicking the "Close Tab" option in a context menu (the *context menu* method)

We evaluated these methods, modeling the closure of each visible tab in browser windows displaying two to six tabs. The models were implemented in CORE (Constraint-based Optimal Reasoning Engine), a platform for Cognitive Constraint Modeling (CCM)[7]. We chose CORE as our modeling tool because it was readily available, incorporated Fitts' Law [2, 3], used a high resolution cognitive model incorporating eye movements and attention shifts [6], optimized the schedule of user behavior, and included useful visualizations. We knew our initial model predictions would not match user data down to the millisecond; however, they would help us qualitatively compare expert performance for the three methods.

The models predicted that when the target tab is buried (not the active tab), the *tab-X* method would be the fastest. In large part, this was because users had to bring a buried tab to the top prior to closing it with the *corner-X* method. The model data clearly showed different patterns for closure of active (top) tabs: When closing active tabs the *corner-X* method was generally fastest. If most tab closures are top tab closures, then the *corner-X* method would be fastest for expert users. Note that top tab closure would not be faster for novices, who were observed to use the *context menu*.

Communicating Model Predictions

While cognitive modeling is an unfamiliar technique to most software engineers, we found that they respected the method, appreciated the data, and found it useful to inform their decisions. An effective communication tool was the Gantt chart output (see Figure 4). The engineers likened it to their system performance visualizations, and could easily extract information about user efficiency. Looking at these charts it was easy for them to see that the most time-consuming actions were right-hand mouse movements.



This row indicates the actions taken by the right hand across time. The long bars are mouse-movement calculated using Fitts' Law, while the shorter bars are mouse clicks.

Figure 4. A Gantt chart generated from the CORE model for *corner-X* closure, the visualization was provided by CogTool [1]. Each row corresponds to a cognitive, perceptual, or motor resource available to the user (e.g., working memory, vision, right hand). Gray bars indicate the use of resources over time (see [7] for more information).

Collecting performance data

At this point, we sought validation of the model predictions in a small user test. We ran a few Firefox expert users (the authors) through three blocks of 100 of trials wherein they were required to close browser tabs as quickly as possible using each of the three closure methods we modeled (one closure method per block of trials). We created a Firefox extension that would present an alert box in the center of the browser content. When the user clicked OK, the extension would open six tabs in random order and select a random tab on top. We recorded the order of the tab presentation, which tab was closed, the time between when the alert box was closed and the target tab was closed, and the positions of mouse clicks.

The trial data indicated that the models were qualitatively correct in their characterization of the relative speed of tab closure for the three different methods: the *tab-X* method was fastest for buried tabs while the *corner-X* method was fastest for active tabs. We also used the trial data to refine the parameters of the original models so that the models produced millisecond-accurate times for quantitative comparison. The refined models indicated that closing the buried tab with the *tab-X* was faster than the *corner-X* method to an even greater extent than initially predicted. The precise quantitative difference did not directly impact the team's decision as much as confirm that the tab-X would not hinder experienced Firefox users' performance while improving the novice experience.

Modeling Click Accuracy

This user study also raised new questions about the impact of the redesigned UI on user performance. Specifically, the team was concerned that the close icon on the tab would lead to unintended tab closures. In order to select a tab (i.e., view the content of the webpage showing on that tab), it is necessary to click that tab. With an embedded close icon, a user might accidentally close the tab while attempting to select it. To address this issue, we used data from our user study to understand the precision of mouse clicks and to build a simple model of errors (misclicks) during tab closure.

From the user data, we estimated the precision (spatial distribution) of mouse clicks during tab selection. In a simple MS Excel spreadsheet, we used this distribution to generate sets of 300 random click locations for each of several tab sizes. In each case, we recorded the number of clicks that would have fallen outside the tab altogether (miss errors) and the number that would

Predicted error rates based on tab width in pixels

Tab Width (pixels)	Close Errors	Miss Errors
165	1.1%	3.4%
155	1.1%	3.7%
145	1.9%	5.0%
135	2.8%	4.6%
125	4.0%	7.4%
115	5.3%	8.0%
105	6.4%	9.7%
95	8.9%	12.5%
85	10.3%	16.6%
75	12.8%	22.6%

Table 1. For tab widths from 165 pixels to 75 pixels, we predicted errors rates for both accidental closure with *tab-X*, and the likelihood a user would miss the tab target.

have fallen on the embedded close icon (closure errors). This model assumes that the user will not adjust (either in speed or in targeting of clicks) to the presence of the embedded close icon. Thus, the model should slightly overestimate error clicks.

Error rates derived from the model are shown in Table 1. The model predictions indicated that close errors would rarely occur with reasonably-sized tabs. Indeed, the model predicted that miss errors should occur at a higher rate than close errors. This information put the team at ease with respect to the accidental closure issue and helped them make a more informed decision about the minimum width allowed for a tab with an embedded close icon.

Overall, our cognitive modeling work suggested that the *tab-X* design was unlikely to impair expert performance (and might even enhance it). This information, in combination with the knowledge that tab closure with the *tab-X* method was easily discovered by novices, made a strong argument for the adoption of the *tab-X* design.

What about real users?

In the end, we conducted a final user study to determine whether novices and experts could close the new tabs. Tab discovery and usage results were in agreement with the earlier results. Seven out of seven novice users discovered tabs and navigated among them. Additionally, all seven novices closed the tabs via *tab-X* without hesitation. Three expert users closed tabs without problems. Only one of those three users even noticed the *tab-X* change. A fourth expert user had previously installed an extension that put his close icon on the tabs. The *tab-X* design did not impact the expert users' performance, nor did we observe any expert users looking for the *corner-X* close icon or moving their mouse to the corner anticipating the *corner-X* close icon.

The final round of user testing gave the team confidence in their new tabbed browsing UI. The final user study also let us investigate things a model cannot predict (e.g., users' ability to discover widgets), and collect qualitative information about user satisfaction and perceived ease of use. These data were important to a team with an existing user base.

Why bother with models?

Often in user research, we conduct small–scale studies to identify usability issues such as discoverability. Combining these studies with modeling techniques can help one better evaluate UIs by providing a more comprehensive answer to UI questions, providing quantitative measures for comparison, saving time, and enabling better-targeted usability studies.

Comprehensive analysis

The combination of qualitative user test data and quantitative modeling results proved to be a far more convincing package of evidence than either result in isolation, given the variety of perspectives in the design and development team. The data were compelling: they explained the results of user tests, accommodated a variety of concerns (including the experience for novice and expert users), and provided performance measures including speed and error rates. This combined approach gave the team confidence in the decision that they made.

Data-driven comparisons

Models support the quick evaluation of UI alternatives based on expert performance, rather than based on heuristics, experience, opinion or conjecture. Model data appeal to designers and engineers because the data can help them quantify performance for comparison. Model software can also produce powerful visualizations (e.g., Gantt charts) that facilitate communication with product managers, developers, and engineers.

Save time in the long run

Cognitive modeling with CORE did require an investment of time and effort up front. The modeler was an expert user of CORE and a member of that platform's development team. However, once our initial models were completed, obtaining further modeling data required a minimal investment. We found that modeling let us quickly produce informed, specific recommendations regarding alternative UIs. Modeling saved us the time and money that would have been spent recruiting users or conducting studies to evaluate alternative UIs for expert performance. Overall, we found that modeling was a flexible and economical choice in comparison to conducting a full-scale user study.

Better target your usability studies

This approach also let us focus our studies on usability concerns outside the scope of the models themselves such as novice behavior, learnability, discoverability, and qualitative measures of satisfaction. Additionally, modeling can reduce the need to run multiple UIs in a study for comparison, or replace some benchmarking-style studies.

The combination of small usability studies and cognitive modeling helped the development team make an informed decision about critical aspects of their UI. By studying both novices and experts along with cognitive modeling, we were able to better understand the novice experience, routine work, and expert performance. This approach was important for a browser used daily by millions.

Acknowledgements

Thanks to Ben Goodger and Brian Rakowski of Google, as well as Mike Beltzner of Mozilla for their interest and support. Thanks to Andrew Howes and Robin Jeffries for their comments on drafts this paper.

Citations

[1] The CogTool Project, Tools for Cognitive Performance Modeling for Interactive Devices. http://www.cs.cmu.edu/~bej/cogtool/.

[2] Fitts, P.M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology, 47(6)*, 381-391.

[3] MacKenzie, I.S. & Buxton, W. (1992). Extending Fitts' law to two-dimensional tasks. *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, 219-226.

[4] Mozilla Firefox homepage: http://www.mozilla.com/firefox/.

[5] Palmer, S.E. (1992). Common region: A new principle of perceptual grouping. *Cognitive Psychology*, 24, 436-447.

[6] Salvucci, D. D. (2001). An integrated model of eye movements and visual encoding. *Cognitive Systems Research*, 1(4), 201-220.

[7] Vera, A., Howes, A., McCurdy, M., & Lewis, R.L. (2004). A constraint satisfaction approach to predicting skilled interactive performance. *Proc. of the SIGCHI Conference on Human Factors in Computing Systems.*