

NOV 05 1996

OSTI

A Zooming Web Browser

Benjamin B. Bederson, James D. Hollan
Jason Stewart, David Rogers, David Vick
Computer Science Department
University of New Mexico
Albuquerque, NM 87131
(505) 277-3112

(bederson, hollan, jasons, drogers, dvick)@cs.unm.edu

Laura Ring, Eric Grose, Chris Forsythe
Statistics and Human Factors, MS-082
Sandia National Laboratories
Albuquerque, NM 87185
(ltring, emgrose, jcforsy)@sandia.gov

ABSTRACT

We are developing a prototype zooming World-Wide Web (WWW) browser within Pad++, a multiscale graphical environment. Instead of having a single page visible at a time, multiple pages and the links between them are depicted on a large zoomable information surface. Pages are scaled so that the page in focus is clearly readable with connected pages shown at smaller scales to provide context.

We quantitatively compared performance with the Pad++ Web browser and Netscape in several different scenarios. We examined how quickly users could answer questions about a specific Web site designed for this test. Initially we found that subjects answered questions slightly slower with Pad++ than with Netscape.

After analyzing the results of this study, we implemented several changes to the Pad++ Web browser, and repeated one Pad++ condition. After improvements were made to the Pad++ browser, subjects using Pad++ answered questions 23% faster than those using Netscape.

Keywords

World Wide Web, Hypermedia, Information Retrieval, Interaction Technology, Internet, Intranet, Usability, User Studies, Zooming, Visualization, Navigation, Animation, Evaluation, Pad++, Information Display, Human Factors.

INTRODUCTION

The World Wide Web (WWW) is becoming increasingly important for business, education, and entertainment. Popular Web browsers make access to Internet information resources relatively easy for users. Simply by clicking on a link, a new page of information replaces the current one on the screen. Unfortunately, after following a number of links, people can have difficulty remembering where they've been and navigating links they have followed. As one's collection of Web pages grows and as more information of interest populates the Web, effective navigation becomes an issue of fundamental importance.

People with diverse talents, interests, and experiences will be on-line via the Web. Many will be novices with little prior experience using computers. A simple click of the mouse can bring a user from their friend's home page to unknown destinations across the world. This can be

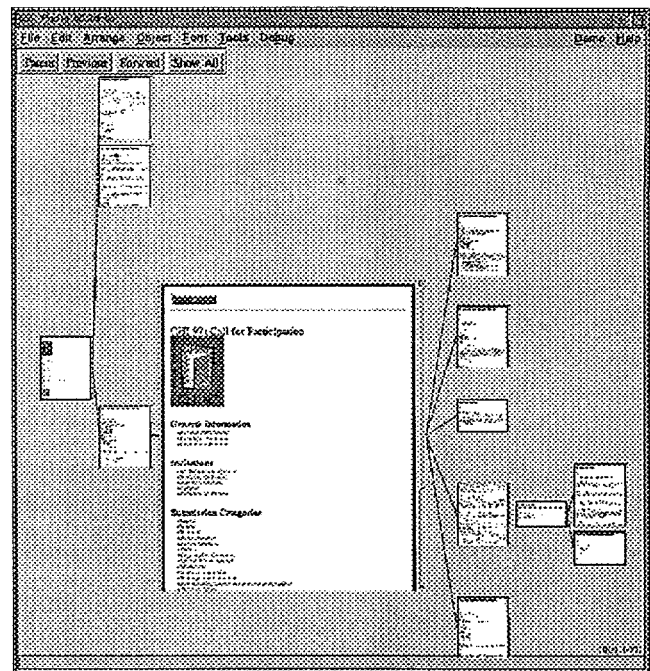


Figure 1: A snapshot of the Pad++ Web browser

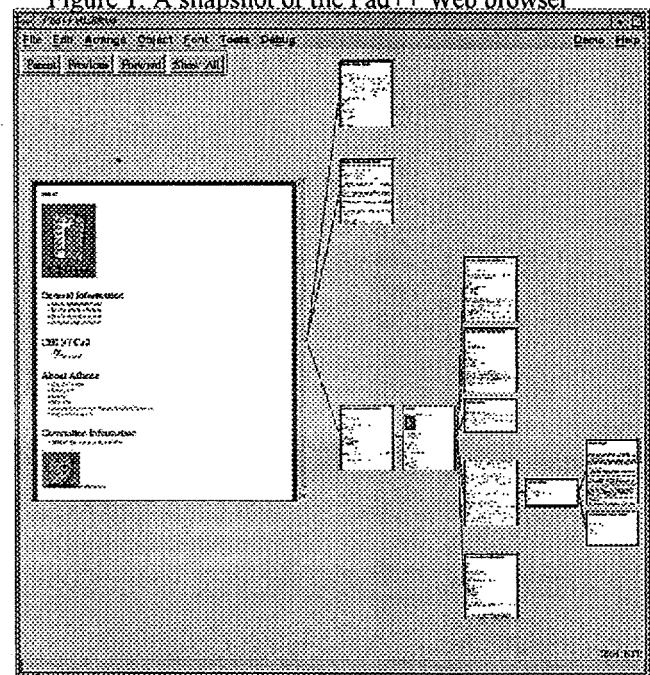


Figure 2: The same Web pages with a different focus

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

disorienting because traditionally, following a cross reference meant shuffling across the library to find another volume. While time-consuming, this reinforced the transition that was taking place. The difficulties that novice users confront can be instructional to developers. While experts may not have as much difficulty, they experience similar cognitive burdens, and may just have a higher threshold before they experience similar difficulties.

While the immediacy of traversing information links offers many advantages, it can also make it difficult to maintain an intuitive sense of where one is, and how one got there. This is a classic problem of hypertext systems. Part of the problem can be attributed to windows-based interfaces. Current window systems don't readily support showing more than a few pages at a time. In addition, each page is usually in a separate window with no depiction of relationships to other windows. Popular WWW browsers, like other applications built according to windowing philosophies, also have this problem, although they do offer some methods to aid navigation by keeping track of interesting sites - usually in hierarchical sets of hotlists or bookmarks.

There are a variety of human factors issues to be considered [9]. Some of these have been addressed by proposed alternatives and extensions to browsers. WebMap is a browser extension that shows a graphical relationship between web pages [5]. Each page is represented by a small circle that can be selected to display the actual page. The links between pages are colored to indicate information about links, such as whether it is a link to a different server or whether the destination page has already been read. These graphs may be saved and used by others. WebBook is a three-dimensional Web browser from Xerox Parc that allows multiple pages to be viewed simultaneously and supports collections of pages in books that replace bookmark files [3].

WWW pages often have links to pages that link back to the starting point. It is difficult to visualize these cycles, and so many Web visualizations are based on hierarchies extracted from a graph of the Web. Some interesting work focuses on alternative visualizations [12]. Furnas [8] shows how multitrees can be used to represent a collection of hierarchies sharing parts of the underlying data. One application of multitrees is visualization of bookmarks from multiple individuals [18].

Another approach to visualizing large information spaces that can be applied to web browsing and navigation involves techniques to show detail at particular nodes while maintaining context. One general approach, fisheye views [7], has been extended with graphics [15], three dimensions [11], hyperbolic representations [10], animation [4], and zooming [1][2][14]. Other techniques include exploiting a large virtual space [6], using lenses or

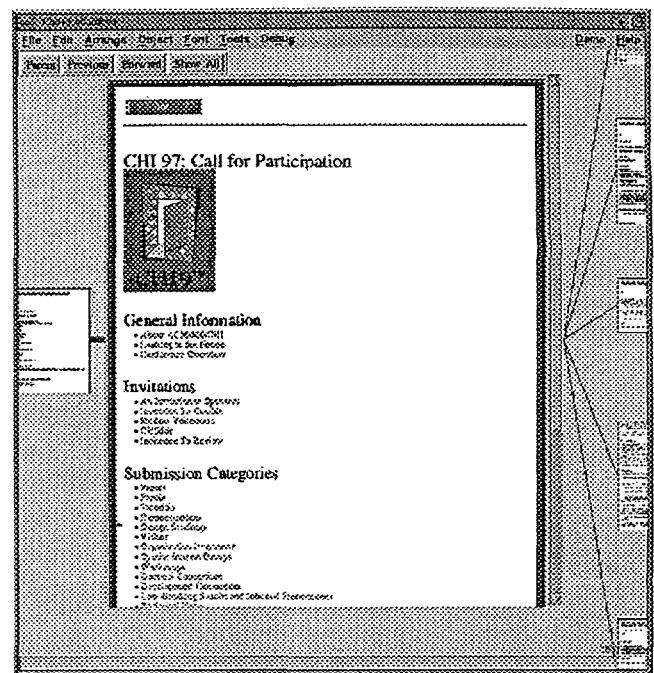


Figure 3: Zoomed into a single page

filters [2][16], and visualizing two dimensional layouts [1][2].

In the sections that follow, we describe our zooming Web browser, discuss user studies comparing it to Netscape, and show improvements we made based on the initial study which resulted in a substantial speedup in the Pad++ Web browser.

A ZOOMING WEB BROWSER

Navigating the WWW presents a struggle between focus and context. As one browses or searches the Web, the need for detailed views of specific items conflicts with the need to maintain a global view of context and history of traversal. The global view is important because it is often necessary to access previously visited pages, either as a starting point to find new information, or for comparison.

We are building a new zoomable Web browser using Pad++, a substrate for building multiscale dynamic user interfaces [1][2]. Pad++ provides an extensive graphical workspace where dynamic objects can be placed at any position and at any scale. Pad++ supports panning and zooming. Zooming can involve simple geometric scaling, or what we term *context sensitive rendering*, in which rendering of objects can vary based on factors in addition to scale, such as context of the task or complexity of the information being displayed. Pad++ is built as a component for Tcl/Tk, a scripting language and user-interface library [13].

Pad++ allows Web pages to remain visible at varying scales so the viewer may examine many pages at once. In addition, Pad++ allows the user to zoom in and out,

enabling explicit control of how much context is viewed at any time. To orient themselves, users can simply zoom out to view a number of web pages. To get more detailed views of a particular page, they can zoom in. We think this variable scale contextual display of Web pages can provide important support for navigation. When using our Web browser, users navigate a space filled with familiar objects, not iconified representations of those objects.

Interaction

The user interacts with the Pad++ Web browser in much the same manner as traditional browsers. Clicking on a link loads in a new page.

The "forward" and "back" buttons from traditional Web browsers are slightly modified to allow access to all the pages previously visited, and not just the ones that are direct ancestors of the current page. Traditionally, the "back" button goes to the page from which the current page was accessed.

We replaced the back button with "previous" and "parent" buttons. The "parent" button does the same thing as the original "back" button, but the "previous" button goes to the previously focused page - whether it was accessed through a link, or panning and zooming.

There is also a button called "show all" which zooms out just enough so all of the pages are visible on the screen. Then, a single click on any page will focus on it, and bring it to a readable size.

While the left mouse button is used for following hyperlinks by pressing on them, it is also used for panning by dragging on the surface. Pressing on the middle button zooms in until the button is released. Pressing on the right button zooms out until it is released. Pad++ zooms around the mouse cursor, so the user can move the mouse while zooming to control which point the view is zoomed around.

Our Web browser combines a basic focus-driven layout with automatic zooming and panning to support navigation. The software allows the user to select a focus page. That selection animates the page to occupy a larger section of the display. Pages farther from the focus page get increasingly smaller, resulting in a graphical fisheye view. See Figures 1 and 2 for snapshots of the Pad++ Web browser with different pages focused upon where the view is zoomed out so many pages can be seen. Figure 3 shows a zoomed in view where a single page may be read. Ancestor and descendant pages are visible on the sides, and can be clicked on to bring them into focus.

Clicking on a link brings up a new page, adds it to the tree of pages, and causes the tree to restructure itself. Unlike other Web browsers that immediately replace the current page with a new page, the restructuring process is animated so that users can understand how the tree is

being. The animation helps maintain object constancy and the graphical depiction of links highlights relationships between pages. The new page becomes the current focus and is moved to the center of the screen at a size suited for viewing. The user may designate any existing page to be the current focus by clicking on it.

Clicking on a link to a page that has already been loaded focuses on that page. Even if it was originally accessed from another part of the tree, its position within the tree is not changed.

As in earlier fisheye displays, our basic layout function assigns a degree-of-interest value to each node in the tree based on its distance from the focus page. We define the distance to be the shortest path between two pages. This value is then used to determine the size of each node. Because long pages have such a big effect on the layout, we added scrollbars to pages longer than a given size. This way, you can still see the structure of the pages, but long pages aren't as disruptive.

USABILITY TESTING

We hypothesize that the Pad++ Web browser will improve users' abilities to retrieve, comprehend, and integrate data from the WWW. We think that the ability to see multiple pages simultaneously, and to learn the structure of the Web will facilitate tasks of this nature. We chose to study the Pad++ Web browser in a focused technical domain to make it easier to evaluate, by allowing measurement of times to answer simple questions. In addition, this domain is representative of the use of a Web browser within a product development environment.

The Stimuli

The usability testing was designed to test how well the Pad++ Web browser could perform as a viewer for this type of database. More specifically, we created a visual database of Web pages. We then compared how long it took to access information from these pages using the two browsers, Pad++ and Netscape.

We designed a hypothetical database containing the evolution of a manufacturing process over a series of six versions. The entire stimuli set consists of 31 Web pages (Figure 4). The stimuli contains an *Overall* page which contains information on the database and links to the *Design and Manufacturing Process* pages and links to the *Budget and Schedule* pages for each of the six versions. Six *Budget and Schedule* pages (Figure 5) give an overall estimate of time and the budget for each version. Six *Design and Manufacturing Process* pages contain the process flowcharts for each version (Figure 6). The process flowcharts are WWW image maps which link to notes about the process. The final 18 pages contain these notes.

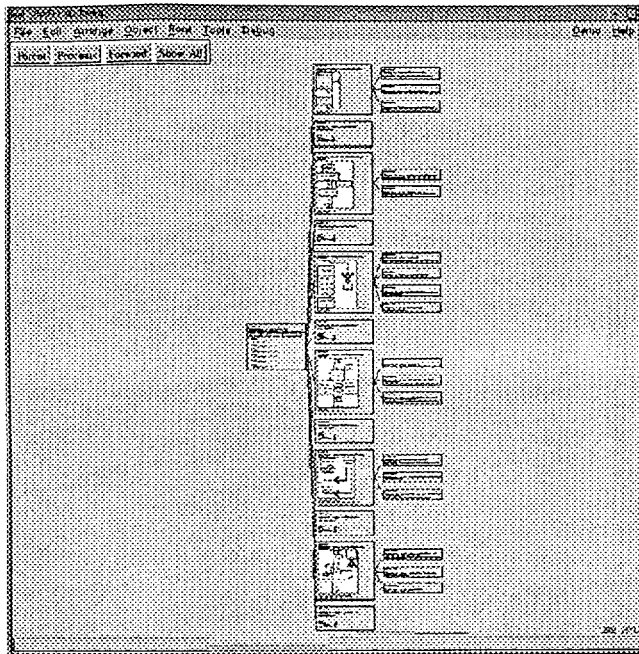


Figure 4: Overview of entire stimuli

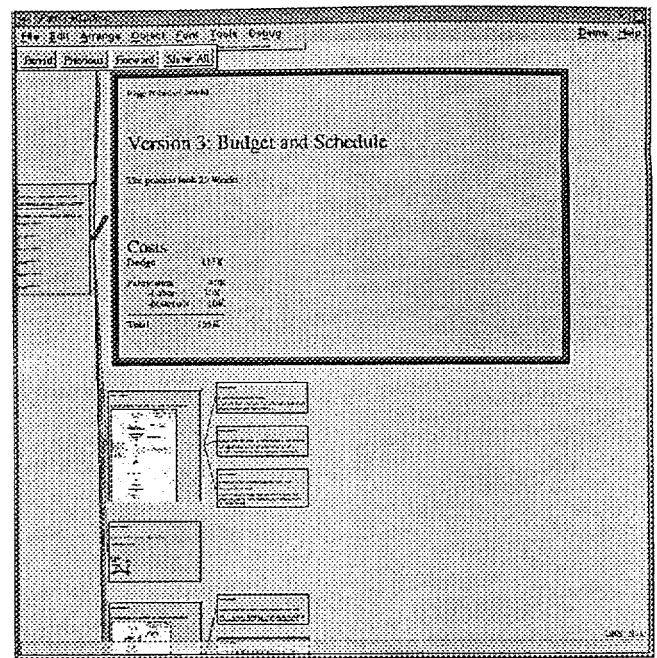


Figure 5: A Budget and Schedule page from stimuli

Task 1 required the subjects to navigate to specified pages in the data base and enter a page number found on the top of each page. For example, the subjects were asked to enter the page number of Version 5's *Budget and Schedule* page. This tested speed of navigation within the browser.

Task 2 and Task 4 tested the users ability to retrieve and comprehend information about the whole database. In Task 2 the subjects were asked to count the number of versions with certain features or where certain events occurred. For example, the subjects were asked to indicate the number of versions that had a new step added. Task 4 required the user to identify versions with certain features or where certain events occurred. For example, the subjects were asked which version had a problem with fabrication.

Task 3 tested the users ability to retrieve and integrate information found within different pages of the database. The subjects were asked to compare different versions. For example, the subjects were asked to indicate which version, out of versions one, two and three had the highest material cost. This required the subject to look at the budget and schedule page for the three versions and compare each of their material costs.

The Experimental Conditions

Four experimental conditions were examined: *Pad++ Prebuilt*, *Pad++ Interactive*, *Netscape Regular* and *Netscape Graphic*. In the *Pad++ Prebuilt* condition, the entire network of pages was loaded into Pad++ prior to beginning the task. In the *Pad++ Interactive* condition, only the *Overall* page was loaded and the network of pages was built as the user answered the questions. In the

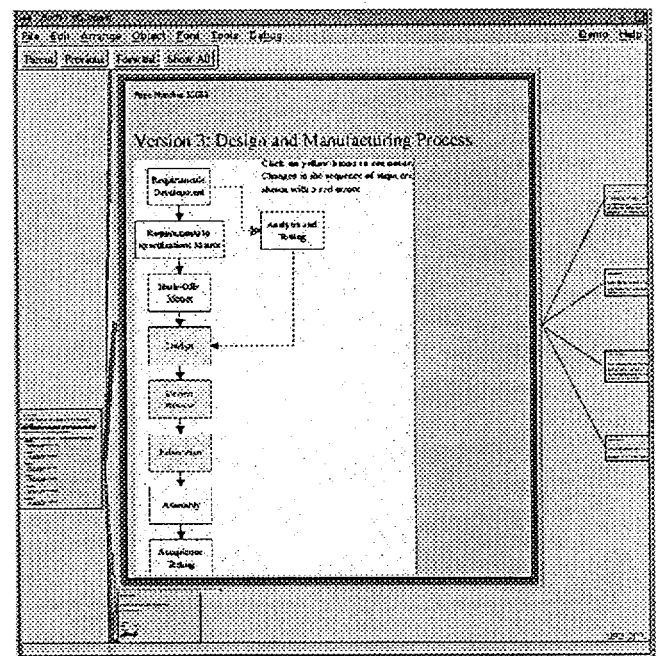


Figure 6: A Design and Manufacturing Process page

Netscape Graphic Condition a WWW image map of the entire database was created by doing a screen capture in Pad++. This condition was added to see if Pad++'s navigational tools caused improvement in performance independent of that caused by the visual representation of the database. The image map used contains a snapshot of the pages depicted in Figure 4.

Training

If a subject reported using Netscape less than 2 hours a week, they were asked to complete a brief training task. The task consisted of browsing through a set series of pages from the University of New Mexico's Web Site. This took approximately 10 minutes. During this task the subjects were instructed on the use of the "forward" and "back" buttons, and how to click on hypertext and image maps in order to follow links.

All but one subject had never used Pad++, although a few had seen brief demonstrations of the software. Three training tasks were used to familiarize the subjects with panning and zooming in Pad++ and using the Pad++ browser. Training tasks 1 and 2 were used to familiarize the subjects with a three-button mouse. In training task 1 the subjects were asked to zoom in and out. In training task 2 the subjects were asked to zoom, pan and select objects. The subjects performed both tasks until they were able to perform 10 operations without error. The third training task asked the subject to open a series of pages within the University of New Mexico's Web Site and then navigate to different open pages. During the training, the subjects were informed of different strategies to use when navigating in the Pad environment.

Subjects

The experiment was carried out at the University of New Mexico. There were 32 participants, but only 30 participants yielded usable data. The two participants whose data was not used had over 50% of the questions incorrect. All of the subjects had completed at least two years of college and all use a computer at least two hours a week. The majority of subjects, 73%, used a computer more than an hour a day. 93 percent reorganized of the subjects had browsed the World Wide Web before, and 80% of subjects browsed the World Wide Web at least an hour a week. 90 percent of the subjects were students and the other 10% were professionals.

Procedures

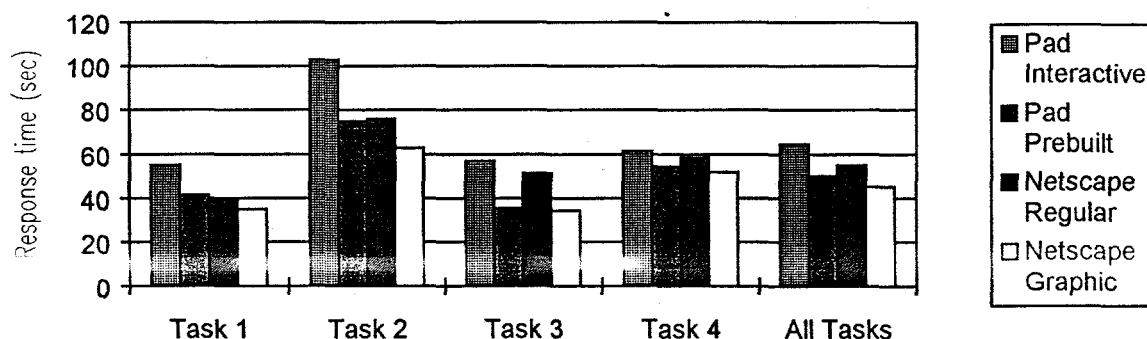
Prior to beginning the experimental tasks, the subjects were trained if needed and then asked to read through an instruction page. The instruction page explained the task and database, and had links to example pages. The subject was instructed to click on all links and look at all the examples. The subjects then carried out trials for the four experimental tasks. In one window, the browser was open with the appropriate initial pages, and in another window, the questions were displayed. A program written in Tcl displayed the questions, and recorded the time spent on each question and the subjects' response. After each question, the entire screen went blank so they could not see the browser and database pages. The subjects were instructed to click on a bar to proceed. This action started the timer and displayed the question and the browser window. Subjects clicked on another bar to enter their response and stop the timer. After the experimental task, a questionnaire was displayed which asked questions about the browser, for information on computer usage and for some demographic information.

Results

In order to determine the quality of the questions, we looked at the percentage of subjects who answered each question correctly. If over 20 percent of the subjects missed the question, it was considered too difficult and thrown out of further analysis. Three questions out of 24 were thrown out for this reason. Two of these questions were in Task 2 and one of these questions was in Task 3.

The dependent variable was the average time it took the subjects to answer each question. Only the times for questions answered correctly were used in calculating the average time. An average time to correct response was calculated first for each of the four tasks. The average times collapsing across tasks were also calculated. The number of correct responses ranged from 16 out of 21 to 21 out of 21 with the majority of subjects having one or

Figure 7: Average Time to Correct Response by Task and Browser Condition after the First Experiment



fewer incorrect responses.

A one-way analysis of variance was performed on each task's average time to correct response. We are aware that in using an alpha of 0.10 for each test, we may be inadvertently increasing our experimentwise Type I error rate, but it is necessary to do so in order to have enough statistical power given the limited number of subjects.

Figure 7 shows a graphical representation of the average time to correct response for each task and collapsed across tasks. The overall F test was only significant for Tasks 1 and 2, using an alpha level of 0.10, $F_{3,26}=2.74$, $p = 0.064$ and $F_{3,25}=2.66$, $p=0.070$ respectively. These results were significant because subjects in the *Pad++ Interactive Condition* took longer to carry out the tasks than did the subjects in the *Netscape Graphic* condition.

These results were largely attributable to technical problems with Pad++ that made it run considerably slower than Netscape. Based on the qualitative data collected by observation and from the subjects responses to the questionnaire, we made various improvements to the Pad++ browser described in the next section.

The speed at which Pad++ was running caused some discontent, but many comments suggested that subjects felt the interface to be intuitive and fun to use. Some of the quotes that indicate the positive aspects of Pad++ follow:

ZOOM is fun!

The fact that one can get the "whole picture" in varying degrees of closeness is fantastic.

I love the zoom in and out. I also liked Show All very much.

The layout of the pages made the task very easy once one figured out that the pages were laid out in the order listed on the parent page. With the color scheme used, the zooming browser allowed answering of some questions merely by panning and examining

the pages from a low resolution view.

Easy to find what I need to know. I did not have to search very much.

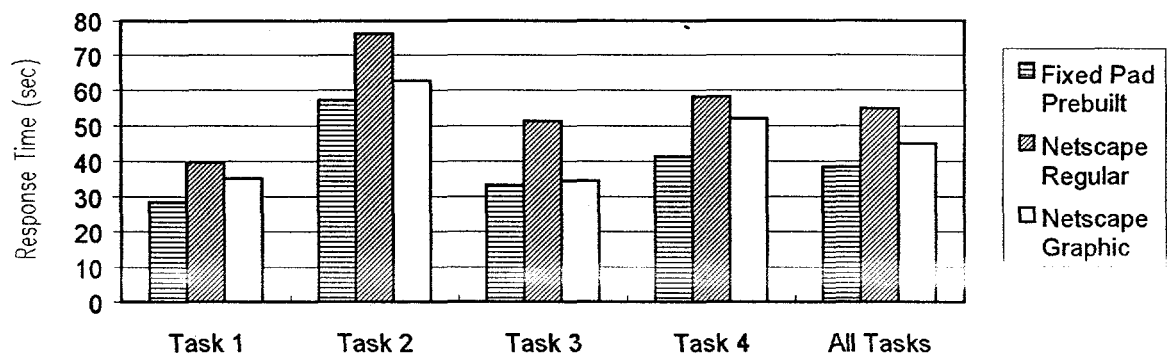
I liked the tree structure, but disliked the occasional delays in updating the display. Once I knew where the pages were in the tree structure, I could quickly find what I wanted.

Another interesting observation made by the experimenter during testing was that subjects varied in their use of panning and zooming available in Pad++. Due to their familiarity with browsing in Netscape, they seemed reluctant to try some of the other features available to them in Pad++. They stuck to what they knew, despite that in training, the experimenter encouraged the subjects to use all available navigational tools. Furthermore, it appeared that there was a positive relationship between subjects use of these navigation features and task performance. In future studies it would be interesting to quantify this difference between subjects and to see if it is positively correlated with task performance.

The Improvements Made to the Pad Browser

- 1) We made Pad++ run roughly twice as fast by using an 8-bit instead of a 16-bit X-server.
- 2) When a page was brought into focus it initially was aligned with the left side of the screen. Some subjects pointed out that this left them feeling disoriented due to a loss of context. We changed this feature so that the page would be centered when it was in focus.
- 3) The fonts were improved from being mediocre to using high-quality system fonts.
- 4) During testing, subjects pointed out that they wanted to be able to defocus a page as well as focus it. A feature was added so the user need only click again on the page to bring it out of focus and make it small again.

Figure 8: Average Time to Correct Response for Improved Pad++ versus the Netscape Conditions by Task



- 5) We made pages thinner so more context could be seen. This was possible because of the improved font quality.
- 6) We fixed a cursor bug which only occurred when we tested the software using novice users. This bug caused the cursor to freeze on a zooming cursor if the user pressed two mouse buttons at once.
- 7) Some subjects indicated that they missed the Netscape feature of the cursor changing when it was over a link. This feature was added in Pad++, so the hand cursor changed into an arrow.

Experiment 2

Following the first experiment, we realized that we needed to make some changes in order to evaluate Pad++ more fairly. First of all, we made the above mentioned improvements. Secondly, since almost all of the subjects in the Netscape conditions were regular Netscape users, we felt that to evaluate Pad++ in a comparable manner, we should use subjects that were familiar with Pad++.

We ran the Pad++ *Prebuilt* condition again with seven Pad++ users using the same procedures as discussed previously. All seven of these subjects worked as part of the Pad++ development team. All were familiar with navigation in the Pad++ environment, and did not need to be trained.

Results

A one-way analysis of variance was performed to compare the new subjects performance to those subjects in the Netscape conditions. The average time to correct responses for each task and averaged across all the tasks are displayed in Figure 8.

The overall F-test approached significance for Task 1 and for the average across all tasks, $F_{2,19}=2.33$, $p=0.124$ and $F_{2,19}=2.43$, $p=0.115$ respectively. Furthermore, tests of the a-priori hypothesis that Pad++ would do better on average than the Netscape conditions yielded a p-value of 0.0599, $F_{1,19}=4.00$ for Task 1 and a p-value of 0.088, $F_{1,19}=3.23$ for the average time to correct response across all tasks.

These results demonstrate that subjects can perform this browsing task 23% faster on average using the Pad++ browser than using Netscape. Many subjects also found the Pad++ browser intuitive and fun to use. In a manufacturing setting, use of the Pad++ browser could increase workers productivity by speeding up the time it takes them to access historical information about a product's process development over time. Furthermore, since using the Pad++ browser allows the user to access such information in a fun and intuitive manner, it may lead to an increased use of historical information with consequent improvements in product and process quality.

CONCLUSION

The World-Wide Web has become an important and widely used resource. Because of this, it is crucially important to address its usability. We have shown one promising technique based on zooming to better support Web navigation. This technique was used to implement a prototype zooming Web browser that was found to be faster at accomplishing specific tasks than a traditional browser.

Pad++ and the zooming Web browser will be made generally available in the near future. To find current information, send mail to pad-info@cs.unm.edu, or look at <<http://www.cs.unm.edu/pad++>>.

ACKNOWLEDGEMENTS

We acknowledge generous support from DARPA's Human-Computer Interaction Initiative (Contract #N66001-94-C-6039). This work also supported by the United States Department of Energy under Contract DE-AC04-96AL85000.

We appreciate the work of our collaborators at the New York University Media Research Lab, especially that of Jon Meyer, in helping us develop Pad++.

REFERENCES

1. Benjamin B. Bederson and James D. Hollan. "Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics", *Proceedings of UIST'94* ACM Press, 17-26.
2. Benjamin B. Bederson, James D. Hollan, Ken Perlin, Jon Meyer, David Bacon, and George Furnas. "Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics", *Journal of Visual Languages and Computing* (7), 1996, 3-31.
3. Stuart K. Card, George G. Robertson, and William York, "The WebBook and the Web Forager: An Information Workspace for the World-Wide Web", *Proceedings of CHI'96*, ACM Press, 111-117.
4. Bay-Wei Chang and David Ungar. "Animation: From Cartoons to the User Interface", *Proceedings of UIST'93*, ACM Press, 45-55.
5. Peter Doemel, "WebMap - A Graphical Hypertext Navigation Tool", *2nd International Conference on the World-Wide Web*, Chicago, IL, 1994, 785-789.
6. William C. Donelson. "Spatial Management of Information", *Proceedings of SIGGRAPH'78*, ACM Press, 203-209.
7. George W. Furnas. "Generalized Fisheye Views", *Proceedings of CHI'86*. ACM Press, 16-23.
8. George W. Furnas and Jeff Zacks. "Multitrees: Enriching and Reusing Hierarchical Structure", *Proceedings of SIGCHI'94*, ACM Press, 330-336.

9. Wendy A. Kellogg and John T. Richards. "The Human Factors of Information on the Internet", in *Advances in Human Computer Interaction*, Ed. J. Nielsen, Ablex Press, (5), 1-36.
10. John Lamping, Ramana Rao, and Peter Pirolli. "A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies", *Proceedings of CHI'95*, ACM press, 401-408.
11. Jock D. Mackinlay, George G. Robertson, and Stu K. Card. "The Perspective Wall: Detail and Context Smoothly Integrated", *Proceedings of CHI'91*, ACM press, 173-179.
12. Sougata Mukherjea, James D. Foley, and Scott Hudson. "Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views", *Proceedings of CHI'95*, ACM press, 331-337.
13. John K. Ousterhout. *Tcl and the Tk Toolkit*, Addison-Wesley, 1994.
14. Ken Perlin and David Fox. "Pad: An Alternative Approach to the Computer Interface", *Proceedings of SIGGRAPH'93*, ACM Press, 57-64.
15. Monojit Sarkar and Marc H. Brown. "Graphical Fisheye Views", *Communications of the ACM*, 37 (12), December, 1994.
16. Maureen C. Stone, Ken Fishkin, and Eric A. Bier. "The Movable Filter as a User Interface Tool", *Proceedings of SIGCHI'94*, ACM Press.
17. David Ungar and Randy B. Smith. "Self: The Power of Simplicity", *Proceedings of OOPSLA '87*, 227-241.
18. Kent Wittenburg, Duco Das, Will Hill, and Larry Stead. "Group Asynchronous Browsing on the World Wide Web", *Proceedings of the 4th International World Wide Web Conference*, Boston, MA, International WWW Conference.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
