Power Browser: Efficient Web Browsing for PDAs

Orkut Buyukkokten, Hector Garcia Molina, Andreas Paepcke, Terry Winograd
Digital Libraries Lab (InfoLab), Stanford University, Stanford, CA, 94305
{orkut, hector, paepcke, winograd}@cs.stanford.edu,

ABSTRACT

We have designed and implemented new Web browsing facilities to support effective navigation on Personal Digital Assistants (PDAs) with limited capabilities: low bandwidth, small display, and slow CPU. The implementation supports wireless browsing from 3Com's Palm Pilot. An HTTP proxy fetches web pages on the client's behalf and dynamically generates summary views to be transmitted to the client. These summaries represent both the link structure and contents of a set of web pages, using information about link importance. We discuss the architecture, user interface facilities, and the results of comparative performance evaluations. We measured a 45% gain in browsing speed, and a 42% reduction in required pen movements.

Keywords

Web; Browser; PDA (Personal Digital Assistant); PalmPilot; Wireless; HTTP; Proxy; Baby faces.

INTRODUCTION

Mobile access to information is a key to individual productivity. Small handheld computers are becoming more crucial in our daily lives. A handheld device equipped with a browser and a wireless connection provides an opportunity to connect to the Internet at any time from anywhere. Such capabilities will increase the usability of PDAs tremendously by providing access to numerous information services, like travel guides, entertainment advice, latest news, flight schedules, even driving directions.

Unfortunately, small screen size, slow text input facilities, low bandwidth, small storage capacity, limited battery life, and slow CPU speed are serious obstacles to the successful realization of that vision.

Screen size limitations in particular require special attention, because they most directly affect the user's experience. For example, a recent study [12] on the effect of screen size on completing browsing-related tasks shows that users with small screens follow links less frequently than their counterparts who were furnished with larger displays, and that their success rate was lower. The study hypothesizes that this lower success rate is in part caused by the more conservative link exploration behavior of small screen users. The study thus calls for improvements in navigation facilities for small screens.

We expect that palm-sized devices on the Internet will not be used primarily for recreational, undirected browsing, but rather for extracting particular bits of information relevant to a current task. Usually, this process begins somewhere near the correct answer, but involves some amount of navigation to home in on the information target. Navigation thus requires particularly strong support for this 'final approach' phase of information access.

One solution is to use browsers that present Web pages in their full form on small screens. Even with compression, this approach can be problematic because of bandwidth and battery limitations. More importantly, the resulting scrolling requirements in both dimensions tend to be excessive. Horizontal scrolling can be avoided on pages that use ordinary HTML text by formatting to a narrow width. Still, this will increase the page height and force the user to scroll up and down excessively.

Another solution is to provide services that format their content explicitly for small screens. While this is commercially viable and is being widely done, it limits the user to the subset of providers who have prepared material for the PDA (and often who charge for that service).

Instead, our *Power Browser* for palm sized PDAs reflects a complete rethinking of small screen navigation clients. We imposed on ourselves the constraint that no server-side content adjustments are assumed. We analyze and display link structure of browsed pages dynamically, and provide specialized pen-based navigation facilities for exploring that structure. A prototype has been implemented on 3Com's popular Palm Pilot device.

POWER BROWSING

We begin our description with an overview of the architecture in which the Power Browser operates. This is followed by a description of the user's search process while using the tool. After describing several relevant user interface features, we then offer a discussion, and pointers to related work.

Architecture

The size of the Palm Screen is 160x160 pixels in a 6x6 cm area. Because the area is so small, only a small fraction of the data on a typical web page can be displayed at a time. A preprocessing stage is required to select portions of the data to show. The PDA has a processor with the power of a desktop machine in the mid-1980s. Therefore computation-intensive display processing should be performed outside the client as much as possible. More importantly, it is obviously wasteful to download full pages, only to then summarize them at the client, since the data transfer rate is generally low. Instead of downloading pages directly, our client sends requests to a proxy server (see Figure 1).

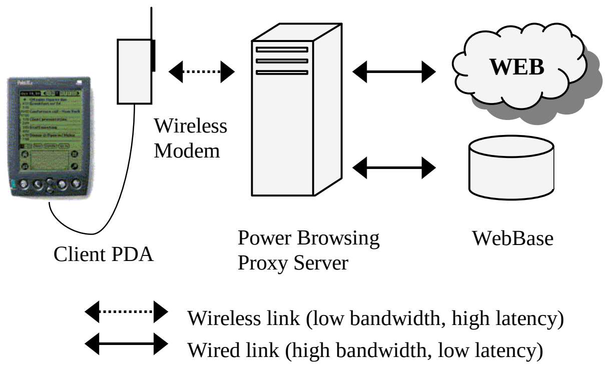

Figure 1: The Power Browser Architecture

The connection between the PDA and the Proxy Server is established through a wireless modem[1]. The server has a wired link to the web, and therefore downloads the pages faster. It processes the data and sends only a small fraction of it to the client at a time. In our experiments, our algorithm for summarization produced a reduction of almost two orders of magnitude (an average 77-fold reduction) in the number of bytes.

The proxy server uses local tools, such as an HTML parser, and an incremental crawler, which can fetch Web pages in the 'neighborhood' of the user's current page. One particularly notable tool makes use of our WebBase facility. WebBase uses an intelligent crawler [6] to collect and store Web pages. It also ranks each page by a 'Page Rank' algorithm [3]. The algorithm ranks a page high if many other pages link to that page. Given a URL for a page that the crawler has already encountered, WebBase provides its rank. When summarizing information for display on the PDA screen, the proxy server can use page rank, if available. See the section on link ordering for more detail.

[1] Currently we use the Metricom Modem. The Palm VII with its built in radio modem was not yet available during the research.

Each user that connects to the Power Browsing Service is identified with his/her user account. This session startup happens transparently to the user. No login process is required. The server opens a new session for each client and maintains browsing information about the user's activities for the duration of the session. The session terminates when the connection between the PDA and the proxy is closed. Whenever a client sends a request, the server returns the cached version of the result, if available.

USER INTERACTION

Starting the Browsing Process

Conventional web browsing is initiated through one of three facilities. The user may manually enter a Uniform Resource Locator (URL). Instead, the user might have a bookmark or find a URL by using a search engine In order to minimize user interaction, the Power Browser presents all of these options in a single initial display (Figure 2).

A URL can be entered on the top lines in the display, using the pen in conjunction with shortcut facilities (the *Cuts* pull-down list) described later. The user's personal bookmarks are displayed in a scrollable list at the bottom of the display. Tapping on one of the entries causes the

Figure 2: Initial Display for Navigation or Search

top lines to be filled in with the corresponding URL. Once a URL is in place, the user taps the Browse button to initiate navigation.

Searches are initiated by entering keywords on the top lines, selecting a search engine from the *With* pull-down list, and tapping the Search button. The keywords are submitted to the search engine via the Power Browsing proxy server. The resulting page is then displayed on the PDA. This feature could be expanded to include results from specialized services, such as stock quotes, driving directions, restaurant, city information or a news service. The search engines that are currently supported by our proxy are Google, AltaVista, Lycos, Infoseek, Yahoo, Northern Light, and Excite.

At the top of the screen is a toolbar with buttons for (from left to right) entering a new bookmark to the bookmark list, deleting a bookmark from the list, getting help, and quitting the application. The bookmarks are stored on the PDA as PalmOS databases (persistent memory objects).

Navigating Through Pages

After the user has tapped the Browse or Search button to initiate browsing or searching, the PDA client sends an appropriate request to the Power Browsing proxy server. Depending on the request, the server either downloads the respective page, or uses a search engine to obtain the search results. The proxy caches all documents received during a user's session. Before delivery to the PDA, the proxy transforms the data into a format appropriate for the PDA's small screen.

Our technique takes advantage of the implicit structure of web pages. Web pages consist of text and multimedia elements, along with embedded links containing URLs for linked pages or files. During the user's final approach navigation phase, our Power Browser displays consists of a set of "link descriptions" which we generate heuristically from anchor text, URL structure, or ALT tags, as appropriate to the link. This structure includes not only the links on a single page, but a hierarchical structure of links on linked pages as well. The user can directly retrieve a page from any link description visible on the screen.

Figure 3 shows how the display uses minimal screen real estate to represent several levels of page structure. Each line of text in Figure 3 is a link description. The descriptions are organized in a tree, similar to the way that file browsers represent nested folders. To conserve screen space, the indentation level is marked by vertical lines rather than by folder or character icons. All text lines in one indentation block represent the links emanating from one page. For example, the page 'Database Group' contains links to pages 'DB', 'Projects', 'Members', and others not shown in Figure 3. In turn, the 'Members' page contains links to 'Andreas Paepcke', 'Andy Kacsmar', etc. Users may expand and collapse the tree through direct manipulation.

One challenge with this approach is the choice of good descriptions for each link. We make this choice heuristically as follows. If the link is associated with text that is, a regular browser would show underlined text to indicate the presence of the link, then we check whether the text is one of a few popular link descriptions that are useless for our purposes. One such 'stop description' is "Click here". If we find that the description is not one of this stop set, then we capitalize the description and use it. This kind of description is sometimes called the 'link anchor'. If we do detect the use of a stop description, we instead turn to the URL associated with the link. If it points to a directory, we use only the right-most element of the URL, and capitalize it. If the URL ends in a file name, we remove the extension, and use the capitalized name.

We experimented with using the titles of pages pointed to by links as the link descriptions. This worked well in that the titles were often good descriptions, but the solution proved to be too expensive. Since we need to generate an entire page worth of link descriptions for each display, the proxy needs to fetch all the corresponding pages from the Web. The consequent increase in latency was too high.

Sometimes, links are associated with images, rather than with text. In this case we look for the alternative text that is sometimes provided for links under images using the HTML "ALT tag". If such alternative text is available for the image link, we use it. Otherwise, we use the URL method described above.

The top of the screen in Figure 3 consists of the command

Figure 3: Tree Control Shows Link Structure

toolbar. The buttons (from left to right) are used for seeing the browse history, making a link the display root, adding a bookmark, and jumping back to the initial screen. The browse history provides an overview of the user's moves through the linked pages (described below). Turning a link into the display root is a tool for cleaning up the display if the user is confident that no backward browsing will be necessary. For example, in Figure 3, if the user did not care to return to any page further back than the 'Members' (of the Database Group) page, then the user could tap on the make-root arrow, and then tap the 'Members' node. Members would move to the top left corner of the screen, removing the indentations on the left.

Underneath the toolbar, there is the title of the root page. This is the initial page that was extracted when the browsing process started. This heading changes only when the user enters a new URL, or uses the 'make root' operation.

Figure 4: Text Display

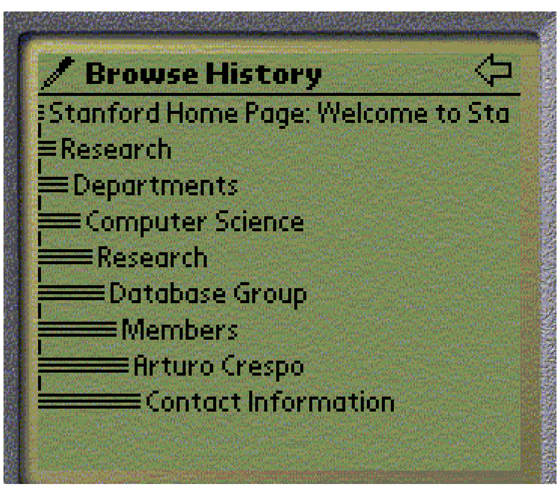

Figure 5: Browsing History

Expanding a node of the tree results in a request to the Power Browser proxy. Users accomplish node expansion through a left-to-right pen gesture over the item to be expanded. Such interactions are described further below. The proxy retrieves the corresponding page from the web, summarizes it by extracting the link descriptions, and returns the result to the PDA for display within the tree control.

This frequent communication could be reduced if we transmitted multiple levels of the tree at once. However, we have found that the 1-2 second delay required for the round-trip communication was worth the decreased bandwidth and PDA storage requirement. The linked structure of pages is transmitted to the PDA gradually, and conservatively, directed by the user's interests as revealed by the requests for expansion.

Viewing Pages

Once the desired page is found, users can view some of its contents. Since the display is low-resolution monochrome, it is usually not worthwhile to display images. Therefore by default, images are ignored. The ALT tag provided in the IMG and MAP environments lets an alternative text string be displayed instead of the image. The ALT text usually gives a reasonable description for the image, however it cannot carry the semantic and visual effect of the image itself. One possible solution to this problem is to display selected images on user demand. In order to display the image on the PDA, a refinement step (i.e., scaling image, decreasing color depth) should be carried out on the server. See [8] for an example of how this can be done.

The Power Browser avoids white space as much as possible. Sequences of paragraphs or line-breaks are collapsed. Many additional text attributes that are visible on standard browsers (color, size, font, alignment, etc.) are ignored as well. On the other hand, structural elements such as line breaks, paragraphs, and tables are used to format the text within the view. Lists are re-formatted into simple text blocks with breaks between successive items. Table rows and columns are folded into text blocks as well. A sample screen shot is shown in Figure 4. The buttons (upper right) let the user move to the top of the text, bottom of the text, or back to the link view, respectively. Underneath the toolbar is the title of the page being displayed. The rest consists of the text itself and a scrollbar.

Example

For explanatory purposes, we present an example here to illustrate how the browser works. Suppose our task is to find contact information for someone named Arturo who is a research member of the Database Group at Stanford University. We call this the *"Arturo task"* for reference. The URL for "Stanford University" can either be entered directly or can be found through a search engine (Figure 2). Once the page at http://www.stanford.edu is displayed on the Power Browser screen, we can expand the tree display repeatedly, thereby following links to *Research*, *Departments*, *Computer Science*, *Research*, *Database Group*, *Members*, and *Arturo Crespo*. This is the state seen in Figure 3. Finally, as we examine the links emanating from Crespo's page, we see a link to *Contact Information* (Figure 3). That is exactly what we are interested in. So instead of expanding the tree further, we look at the page text by tapping on *Contact Information*. This action takes us to the view shown in Figure 4.

The browser also provides an option for viewing the browse history. Notice in Figure 3 that while we can see the title of the root page ("Stanford Home Page: Welcome to Stan"), we cannot see any information about the pages between the root and the Database Group page. Scrolling would reveal this information, but we found that a more powerful display device for browsing history was required. Figure 5 shows the path we followed from the Stanford home page to the desired contact information in compact form. The magnifying glass icon in the tool bar at the top of Figure 3 invokes this overview display.

Finding this information using a traditional full-text browser would have required us to look at all the pages at each level, read through a large amount of text, and try to locate the hypertext that leads to our destination. In this example, instead of looking at eight different web pages, the user can easily navigate through the link structure from the initial page and reach the target. Of course, this approach works for goal-directed tasks, rather than for recreational browsing. It also works best on sites that use sets of links, rather than extensive text and images. For those sites, the browser's strengths come less into play.

Link Ordering

One of the enhancements of the system compared to a traditional browser is the option of ordering the links. A standard browser displays the links in the sequence they appear in the document. Since only a small number of links can be displayed at a time on the PDA, it becomes important to display them in an efficient order. The Power Browser allows users to specify three sorting schemes: original, alphabetical, and page ranked. For instance if we are looking for a person's first name among a large list that is ordered by last name, we would prefer to have the names re-ordered alphabetically according to their first name. This feature was used in our previous example to get the group members ordered alphabetically according to first name.

In other cases, when we see a number of similar links, we might like to visit the ones that are most popular first, to avoid browsing through a large number of pages. This feature is supported using a quality measure for web pages, called Page Rank [3]. Page Rank was first used by the Google search engine to return better search results. A mathematical analysis, calculated on more than a billion hyperlinks on the web is used to estimate the quality or importance of web pages. A page like www.stanford.edu has high importance if, recursively:

1. Many other pages point to www.stanford.edu
2. These pages in turn have high importance.

Other ranking algorithms could, of course, be used for sorting links.

NAVIGATION FACILITIES

Summarization is only one requirement for effective browsing from PDAs. We also need to consider task-specific support facilities that make user tasks easier. We describe the Power Browser's use of the pen as an input device, the use of gestures, animation, and link reduction.

Shortcuts

One of the major differences between a handheld device and a laptop or desktop computer are the input modalities. The PalmPilot does not have a keyboard and most text entry is done using a pen with the Graffiti character set. This makes text entry more difficult, so PDA applications such as the Power Browser try to minimize manual text entry. However, text entry cannot be avoided completely, so the Power Browser provides text shortcuts to. For instance while entering a URL in the initial display, a pull-down menu is provided to insert common prefixes and suffixes (http://, ftp://, www., com., etc.). Commands for cut, copy and paste are supported in the standard manner as well.

Gestures

Another key difference between PDAs and desktop machines is the PDA's lack of a mouse. The pen carries the functionality not only of the keyboard, but also of the mouse. The pen, while only as versatile in selecting as a one-button mouse, does allow us to introduce gestures into the user interface.

Gestures are limited size and duration pen-tip trajectories (strokes) of distinguishable shapes. We can save screen real estate by using gestures in place of buttons to invoke actions. The use of gestures can potentially interfere with text recognition, but this problem is avoided on the Palm Pilot, by performing text recognition only in a dedicated portion of the screen, below the information display. The display area itself is therefore available for gestures.

The tree control used for displaying the link structure (Figure 3) is operated using gestures. A node is expanded with a left-to-right gesture over the link to be expanded, and collapsed with a right-to-left gesture. Up-down and down-up gestures operate scrolling. The text of a node's associated page is displayed by a single pen-tap on the link description.

Both position-dependent and independent gestures are used in the text view (Figure 4). Here, a right-to-left gesture carries out the same action as a back button, returning the user to the previous view. Any other pen trajectory results in selecting a region of the text to enable copying/pasting operations.

Animation

Because of the limited screen size, it is essential to move the tree control up and down as the structure is modified to keep the most relevant area in view. When a tree node is expanded, the selected node moves to the top to make it possible to view as many new nodes as possible Whenever after a node collapses, the bottom lines of the view may become empty, the tree structure is moved down.

We initially recalculated the screen and displayed the result immediately. When users tested the system, the sudden redisplay proved confusing. They found it difficult to get reoriented when the screen was redrawn, because nodes in the tree might move up or down on the display, or new nodes may be introduced. We therefore added animation. Instead of moving nodes abruptly, nodes that change their positions are scrolled to their new location. The scrolling speed is a key parameter in the animation. An animation that is too slow increases response time and is unpleasant. On the other hand if the speed is too fast, the eyes can't trace the rapid movement and the movement can result in flickering on the screen. We determined an acceptable speed experimentally. The addition of animation had a large positive impact on usability.

Seeing the Forest AND the Trees

Establishing both overviews and a notion of location throughout navigation is important for successful browsing. Conventional browsers provide buttons to move one step forward or backward, and a linear menu display of the link path leading to the current page. This makes it difficult for the user to establish a sense of location since the link structure of the web is a graph that can be traversed arbitrary order.

For instance, while following links, the user may hit upon the same page multiple times, along varying paths. In this case, backtracking from that same page will return to a different page each time. The Power Browser's tree arrangement of links displays both the user's location and the neighboring navigation environment at all times. One positive consequence is that users can jump to a sibling page with a single action. (Siblings are pages that are pointed to by the same parent page). In conventional browsing, the user has to backtrack and move forward again to reach siblings.

Link Redundancy Reduction

Another method for navigation support is hiding; restricting the navigation space by hiding links to pages. Most web sites have many duplicate links within their link structure. For instance, some pages provide a navigation bar support that gives direct access to the most critical web pages within the site. This structure is often repeated on each page. Also, many pages provide a link to their parent page or all the way back to the root page of the site. There can be duplicate links within the same page as well. Some pages provide alternative options for the same links (e.g., a page accessible through both the anchor and an image). The proxy server removes any duplicate links that it finds. This assures that each link on the tree control is unique. If the same link has multiple descriptions, a heuristic is used to choose the best one. Link descriptions are preferred to alternative text. If there are multiple link descriptions, the longest one is chosen. When we tested the browser with duplicate removal, we observed that navigation became easier since it reduced the cognitive load for the user to recognize duplicate links. Restricting the navigation space compacted the link structure and reduced the complexity of the unrestricted space.

TESTING

We performed three kinds of experiments to test the validity of our approach. The first measured system performance, the second measured the best possible user-level performance for the *Arturo task*, and the third measured actual user performance for the *Arturo task* and five others.

System Performance

One major design decision was the use of a proxy. Proxy-based designs carry intrinsic disadvantages: a proxy must be available at all times; for reasons of scalability, proxies need to be replicated when large numbers of clients are to be served, and they add an additional hop on the network. We decided on this solution nevertheless, in order to conserve bandwidth and CPU/battery activity on the PDA. Careful design was invested in the communication protocol between PDA and proxy. The size of data packets sent between the client and the proxy server was minimized.

First, there is the proxy-side reduction from the full Web page to a list of link descriptions and their associated URLs. Measured for a relatively small list of random Web pages, this yielded a factor of about 20 in byte size reduction. This number obviously varies with the composition of the pages. Rather than sending the link descriptions and URLs to the client, the proxy assigns an object identifier (OID) to each URL. These OIDs are much shorter than the URLs themselves, and added another factor of 3.5 in byte size reduction. What is transmitted to the PDA is the resulting list of link descriptions, and the associated OIDs. The overall savings is a roughly 70-fold reduction in the number of bytes. When the user performs an action on the tree that requires proxy activity, only the affected OID and the action are transmitted to the proxy.

Lower Bound for User Performance

In order to find and compare the best possible user-level performance, we measured the minimal amount of pen activity necessary for users to perform the *Arturo task* once they knew exactly what pages and information is required, and the best possible time to run through the task. We performed this analysis on the Power Browser and on three other PDA-based browsers: ProxyWeb [16], PalmScape [13], and HandWeb [17]. All of these other browsers attempt to display Web pages as similarly as possible to what a full-sized browser would show.

Table 1 summarizes the results. The total number of pen moves is the sum of pen taps needed for scrolling, the taps necessary for selection, and gestures. The time for task completion was the result of running through the task as quickly as possible, once the necessary link sequence was known to the operator. The completion time excludes the time for entering the initial URL, but it includes all connection time through the wireless network, including the initial connection setup. We ensured that no pages had been previously cached. The tests were run with a Palm IIIx, connected with a Metricom Ricochet modem with nominal speed of about 19kb/s.

	Pen-taps for Scrolling	Pen-taps for Selection	Pen Gestures	Pen Moves	Total Time (secs)
Power Browser	3	1	8	12	80
ProxiWeb	21	8	N/a	29	170
PalmScape	13	8	N/a	21	234
HandWeb	22	16	N/a	38	254

Table 1: Comparing optimally possible performance for the *Arturo task*

As we can see from the table, there was a reduction by an average factor of 2.74 of task completion time and a factor of 2.4 in the number of required user interactions on the Power Browser. For the slower browsers the numbers are even stronger: a factor of more than 3 in completion time reduction. With slower links, the reduction would be even more significant. Similar differences in performance were obtained for the tasks that are described next.

Actual User Performance

In order to measure actual user performance, we had 10 users perform a total of six tasks. Each user performed three tasks on the Power Browser, and the other three tasks on ProxiWeb. We varied the sequence in which subjects were exposed to the two browsers. We chose ProxiWeb for the comparison, because it had the best performance bound as per Table 1. The tasks were as follows:

Task 1: Beginning at http://www.mit.edu, find the date of the first day of classes in the fall semester of the '99/'00 academic year.

Task 2: Use the result of a Google search for "CHI 2000" to find the panel co-chairs for the CHI 2000 conference.

***Task 3*:** Beginning with Google search "New York Public Library", find the Manhattan branch's opening hours.

Task 4: the *Arturo task*.

Task 5: Beginning with Google search "Metronome ballroom", find the price of group dances for *Gold International Style*.

Task 6: Beginning with http://www.usatoday.com, find the NFL league TV schedule.

All subjects were Stanford Physics, Psychology, and Computer Science graduate students who use computers at least 3 hrs/day, and perform at least one Web search in a normal working day.

Chart 1 shows the completion times for the tasks. The average time savings across all tasks and subjects was 45%. Chart 2 shows the number of pen actions performed by the subjects for each task, and the subset of these

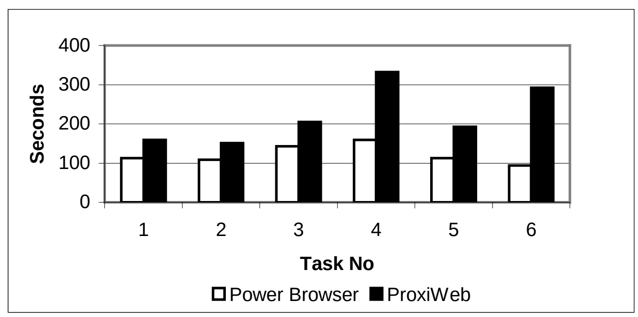

Chart 1: Average task completion times

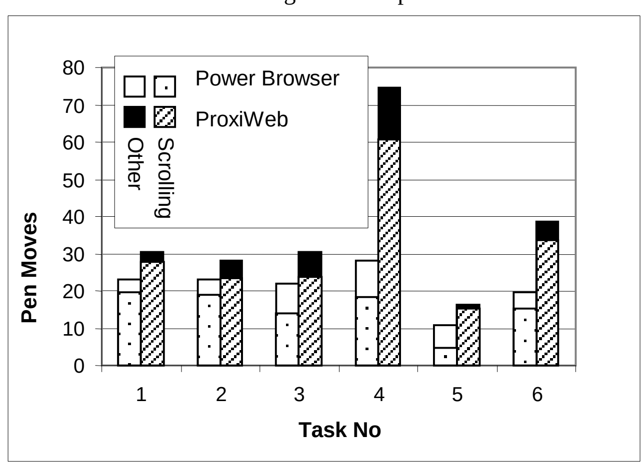

Chart 2: Average pen taps/gestures, and scrolls

moves that were required for scrolling. The ‘others’ category comprises pen taps and gestures. The average pen action savings afforded by the Power Browser was 42%.

When using the Power Browser for the *Arturo task*, subjects took about double the best possible attainable time (Table 1), and they performed about twice as many pen moves as the absolute minimum. These same ratios held for task 6 for which we generated the lower bound measures as well.

RELATED WORK

Browsing the WWW from PDAs has been demonstrated in research projects and in commercial applications [2, 10, 9, 16, 17, 13]. One related product is the Pocket Internet Explorer by Microsoft for Windows CE devices. Browsers for the PalmPilot include Topgun Wingman/ProxiWeb [9,16], HandWeb [17] and PalmScape [13]. These browsers attempt to render content as fully as possible, and do not provide any additional features to assist navigation. Wingman/ProxiWeb uses a proxy server whereas HandWeb and PalmScape download the pages directly.

Much research has been done on using the hypertext structure of the web to improve navigation and to build useful applications. Bachiochi [1] added navigation buttons to a browser that enable maneuvering within hierarchical Web site structures, based on one’s current position. The ParaSite system [18] exploits the link information on the Web to find moved pages and un-indexed information. Scratchpad [15] proposes a set of mechanisms based on breath-first traversal of web pages. Nif-T-nav [11] provides a hierarchical navigator and shows the state of the navigation using a tree structure. Similarly, WebToc [14] visualizes the contents of a Web site with a hierarchical table of contents. Brusilovsky [4] describes Adaptive Navigation Support (ANS) methods for Web-based systems, like link ordering. WebMap [7] creates a graphical map showing the navigation history. Cha-cha [5] uses a pre-computed tree arrangement to display results of site searches.

CONCLUSION

In summary, we have described our Power Browser for PDAs. It can significantly reduce browsing times for directed tasks. The browser achieves its gains by intelligently presenting the link structure of Web pages, and by providing effective and natural commands for navigating this structure.

REFERENCES

1. Bachiochi, D., Berstene, M., Chouinard, E., Conlan, N., Danchak, M., Furey, T., Neligon, C. and Way, D., Usability Studies And Designing Navigational Aids For the World Wide Web, in the Proceedings of the 6th WWW Conference, 511-517, 1997.
2. Bickmore, W., Timothy, Schilit, N., Bill, Digestor: Device-independent Access to the WWW, in the Proceedings of the 6th WWW Conference, 1997.
3. Brin, S. and Lawrence P., The anatomy of a large-scale Web search engine, in Proceedings of 7th International WWW Conference, 101-117, 1999.
4. Brusilovsky, P., Eklund, J., Schwarz, E., Adaptive Navigation Support in Educational Hypermedia on the World Wide Web, in the Proceedings of INTERACT, 278-185, 1997.
5. Chen, M., Hearst, A., Marti A., Hong, J., and Lin J., Cha-Cha: A System for Organizing Intranet Results, in Proceedings of the 2nd USENIX Symposium on Internet Technologies and Systems. Boulder, CO. Oct. 1999. To Appear.
6. Cho, J., Garcia-Molina H., Page L., Efficient Crawling Through URL Ordering, in the Proceedings of the 7th International WWW Conference, 1998.
7. Domel, P., WebMap: a graphical hypertext navigation tool, in the Proceedings of the 2nd WWW Conference, 1994.
8. Fox, A., Brewer, A., E., Reducing WWW latency and bandwidth requirements by real-time distillation, in the proceedings of 5th WWW Conference, 1445-1456, 1996.
9. Fox, A., Goldberg I., Gribble, S. D., Lee, D. C., Polito, A. and Brewe, E. A. Brewer, Experience With Top Gun Wingman: A Proxy-Based Graphical Web Browser for the 3Com PalmPilot in the Conference Reports of Middleware, 1998.
10. Gessler, S. and Kotulla, A. PDAs as mobile WWW browsers, in Proceedings of 2nd International WWW Conference, 53-59, 1994.
11. Jones, L., Krisen, nif-T-nav: A hierarchical navigator for WWW pages, in the Proceedings of the 5th WWW Conference, 1996.
12. Kawachiya, K. and Ishikawa, H. Improving Web interaction on Small Displays, in Proceedings of 8th International WWW Conference, 51-59, 1999.
13. Kazuho, O., PalmScape: http://palmscape.ilinx.co.jp/
14. Nation, A. D., Plaisant, C., Marchionini, G. and Komlodi, A., Visualizing websites using a hierarchical table of contents browser: WebToc, in the Proceedings of the 3rd Conference on Human Factors and the Web, 1997.
15. Newfield, D., Sethi, S., B. and Ryall, K., Scratchpad: Mechanisms for Better Navigation in Directed Web Searching, in the Proceedings of UIST, 1998.
16. ProxiNet, Inc., ProxiWeb: http://www.proxinet.com/
17. Smartcode Software, HandWeb: http://www.smartcodesoft.com/products/handhelds/hh_handweb.html
18. Spertus, E., Parasite: Mining Structural Information on the Web, in the Proceedings of the 6th WWW Conference, 201-211, 1997.