What Did They Do? Understanding Clickstreams with the WebQuilt Visualization System

Sarah J. Waterson, Jason I. Hong, Tim Sohn, James A. Landay
University of California, Berkeley
527 Soda Hall
Berkeley, CA 94720-1776
+1 (510) 642-3437
{waterson, jasonh, tsohn, landay}
@cs.berkeley.edu

Jeffrey Heer
Xerox PARC
3333 Coyote Hill Road
Palo Alto, CA 94304
+1 (650) 812-4373
jheer@parc.xerox.com

Tara Matthews
Seattle University
900 Broadway
Seattle, WA 98122
+1 (206) 296-5510
matthewt@seattleu.edu

ABSTRACT
This paper describes the visual analysis tool WebQuilt, a web usability logging and visualization system that helps web design teams record and analyze usability tests. The logging portion of WebQuilt unobtrusively gathers clickstream data as users complete specified tasks. This data is then aggregated and presented as an interactive graph, where nodes of the graph are images of the web pages visited, and arrows are the transitions between pages. To aid analysis of the gathered usability test data, the WebQuilt visualization provides filtering capabilities and semantic zooming, allowing the designer to understand the test results at the gestalt view of the entire graph, and then drill down to sub-paths and single pages. The visualization highlights important usability issues, such as pages where users spent a lot of time, pages where users get off track during the task, navigation patterns, and exit pages, all within the context of a specific task. WebQuilt is designed to conduct remote usability testing on a variety of Internet-enabled devices and provide a way to identify potential usability problems when the tester cannot be present to observe and record user actions.

Categories and Subject Descriptors
H.5.4 [**Information Interfaces and Presentation**]: Hypertext and Hypermedia – *user issues.*

General Terms
Measurement, Design, Human Factors.

Keywords
Usability Evaluation, Log File Analysis, Web Visualization, Semantic Zooming, Remote Usability Evaluation.

1. INTRODUCTION
Web site usability remains a serious problem today. Studies have shown many web users find it difficult to locate products they want to purchase and often give up [19, 25]. The usability of a web site is one key factor to its success, particularly for increasing sales and repeat business on e-commerce sites [17, 20]. However, gathering and analyzing the appropriate usability data is a complicated matter. Traditionally, usability testing and data gathering is done by bringing users into usability labs and observing their actions as they perform certain tasks. This provides important qualitative data, but is expensive, time consuming and tends to reflect only a small number of users. Analysis of server logs is frequently used to quantitatively understand what large numbers of users are doing on a site, but it is difficult to comprehend the actions, goals, and motivations of individual users [5, 8].

In the past few years, a number of online survey and logging tools [16, 22] have been developed in an attempt to quickly gather not only more data than traditional lab style testing, but also higher quality data than that provided via server log analysis. These tools, referred to here as remote usability testing tools, recruit users to perform specific tasks on web sites from wherever the user happens to use the Internet. Qualitative data and demographics are gathered using online questionnaires. The participants' clickstreams are logged as they perform the task. A clickstream records the path the user took through the web site, along with the time he or she spent on each page, and any data they entered and received.

Clickstreams may be gathered by a variety of methods. However, finding a good tool to aggregate, interpret, and analyze the data remains difficult. In order for the clickstream data gathering tools to be successful, sophisticated analysis methods need to be provided to answer usability questions deeper than which pages received the most hits or how may shopping carts are abandoned. While these factors are important pieces for understanding the user's experience on the web site, they must be studied in the context of the task and the users intentions. This is especially important for situations where human observers are not present.

In this paper we discuss our research into the visualization of remote web usability clickstream data generated by WebQuilt, a proxy-based usability logging system [11]. WebQuilt can unobtrusively gather tasked-based clickstreams from a variety of internet-enabled devices such as PDAs, WAP phones, and desktop computers. Designed to support existing online survey and participant recruitment tools, the WebQuilt logging system can also act as its own stand alone remote usability testing and analysis system. A web designer would first set up several tasks. and recruit participants, perhaps via email, to perform these tasks. As the tasks are performed, the WebQuilt proxy logger

automatically collects the data. Demographic, pre- and post-task questionnaires, as well as instructions or task descriptions can be incorporated into the test design as well. The rest of this paper will discuss related work, the WebQuilt visual analysis tool, its implementation, and future research directions.

2. RELATED WORK

A variety of web and Internet related visualization systems have been developed, including ones for displaying an individual user's history in a web browser [24], web site mapping [13], web connectivity [2], information scent [3], and web traffic log data [10, 15]. For a good review of current systems visualizing online activity, see Eick 2001 [7]. Guzdial and colleagues review and introduce several usability log visualizations of desktop-based, non-web systems in [9]. The WebQuilt system differs from these visualization systems in that it is intended specifically for designers to understand task-based usability problems on the web.

A few commercial and research systems for web usability logging and visualization have been developed that share goals similar to WebQuilt. The National Institute of Standards and Technology has developed VISVIP [4], which shows a 3-D representation of individual clickstream paths overlaid on a 2-D graph of the entire web site. While compatible with many operating systems and browsers, the WebVIP logging system requires access to the entire web site topology, as opposed to just the relevant pages. WebQuilt logs only the pages actually visited by test participants and is not restricted to a set of pre-defined pages. Additionally, WebQuilt displays both aggregate and individual paths. Visual Insight's eBizinsights [21] is a sophisticated server log visualization tool for generating custom reports and interactive charts. Designed mostly for management and marketing reports rather than designers, the eBizinsights software is restricted to server-side data gathering, and their Visual Path Analysis visualization only shows traffic flow in and out of a single page at a time.

Vividence ClickStreams [22] displays individual and aggregate user paths through a web site as a hyperbolic tree, using path color and weight to indicate task success and traffic. Blue Martini's ClickViz [1] presents a graph of icons representing web pages and the flow of traffic, with additional controls to modify the display via filtering and sorting. NetRaker Clickstream [16] similarly shows traffic flow between representations of the web pages and uses lenses to distinguish similarities and differences between pages. While these visualization systems all analyze web usability logs, they all vary in the types of data they can gather and present, as well as in the levels of control and interaction with the data. The WebQuilt visual analysis tool expands on the successes of these systems, incorporating some visual elements and interactions similar to ClickStreams and ClickViz, such as path traces representing traffic flow and filtering. WebQuilt introduces semantic zooming for more powerful interaction methods to analyze and interact with the display.

3. CLICKSTREAM VISUALIZATION

The WebQuilt visualization aggregates the data from multiple test participants into an interactive graph, an example of which is shown in Figure 1. Visited web pages are nodes, represented by color blocks in this overview screenshot. The nodes appearing in green are the entry nodes, cyan nodes are web pages where participants exited the task, and the dark gray nodes are neither entry nor exit points. Typically there is only one entry node in a graph, as usability tests start all participants at the same point. Arrows are used to indicate traversed links and when participants hit the back button. Thicker arrows indicate more heavily traversed paths. Arrow color is used to indicate the average amount of time spent before traversing a link, with colors closer to yellow meaning short amounts of time and colors closer to red for longer amounts of time. As an option, a designer designated "optimal" path can be indicated and rendered as a thick blue highlighting along the path (e.g. the top path in Figure 1).

The checkboxes along the bottom indicate which paths are currently displayed, and provides a filtering mechanism for adding or removing individual user traces. On the left-hand side of the interface is a slider used for zooming in and out of the graph to show the web pages and paths at varying levels of detail. Figure 2 shows a zoomed in view of the left-hand side of the graph in Figure 1, including the green start page (the node furthest to the left in the figure). This level of detail, the storyboard level, shows the interaction between a handful of pages. Instead of color blocks, the nodes are now thumbnails of the actual web pages participants viewed, and the details of the path transitions are more distinguishable. Zooming in further to the start page shown in Figure 3, the entire page layout is clearly visible, as is the URL for the page. Additionally, the arrows out of the page are now connected to the links that users actually clicked. Both arrows in and out of the page become translucent at this level, so as not to cover design details in the page below. For any selected links that required scrolling to reach, the resulting out-bound arrows originate from a location on the edge of the image.

Along the top of the page view are two icons. The left pointing arrow icon represents a browser back action. If any user clicked the back button during the task while on the page, a red dot is placed on this arrow button to indicate that action, as shown in the example in Figure 3. The other icon, a computer with a question mark, will have a red dot if the user clicked on a link with a location currently undeterminable by the WebQuilt logging system. For example the locations of form submission buttons and fields are not yet recorded in the system.

Navigation of the graph is currently done by selecting nodes, zooming in and out, and performing navigational gestures on the visualization itself to shift different portions of the graph into the screen display. Gestures, a common interaction technique for pen-based interfaces, allow the user to draw a stroke, such as line from left to right, across the visualization. This left-to-right gesture is interpreted by the system as an action to navigate to the right in the graph. Gestures need not be performed with a pen input device. Holding down the right mouse button and drawing strokes performs the same action. Gestures can also be used for selection of graph elements.

4. VISUAL ANALYSIS

The screenshots in Figures 1, 2, and 3 show an example visualization of ten internet-enabled PDA clickstream traces, where users were asked to use the Edmunds.com PDA web site (http://pda.edmunds.com) to find two pieces of information on the latest Nissan Sentra model. The pages along the blue highlighted path at the top represent the designer-defined path. By looking

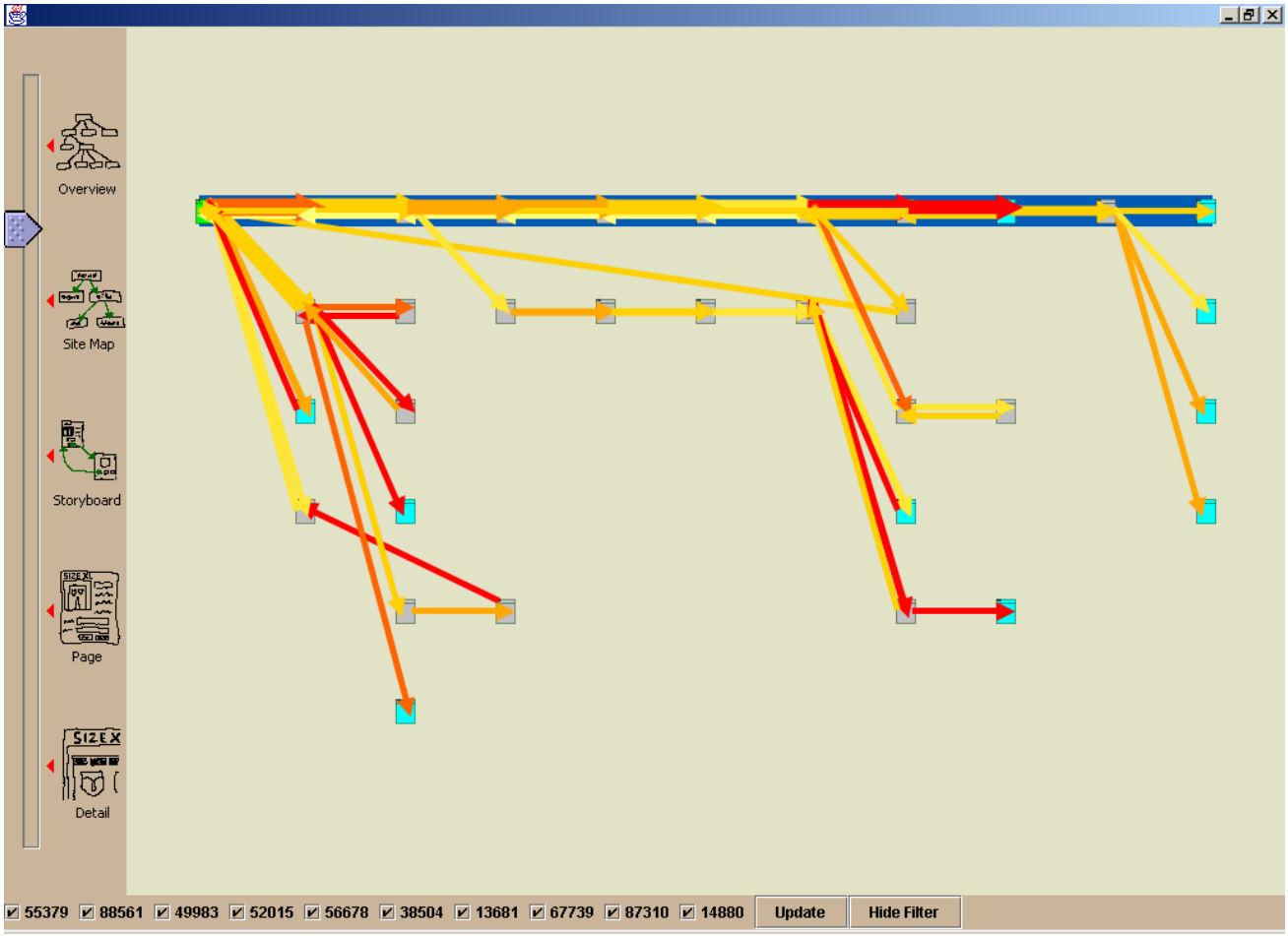

Figure 1: The WebQuilt Visualization. Nodes are visited web pages, and arrows are the traffic between the pages. Entry pages are colored green, and exit pages cyan. Thicker arrows represent heavier traffic. Arrow color is used to indicate time spent on a page before transitioning, where the closer the arrow to red, the longer spent in transition. The designer's path is highlighted in blue. The zoom slider interface along the left hand side is used to change the zoom level. The checkboxes along the bottom indicate which participant paths are currently displayed, and can be used to add and remove paths from the display.

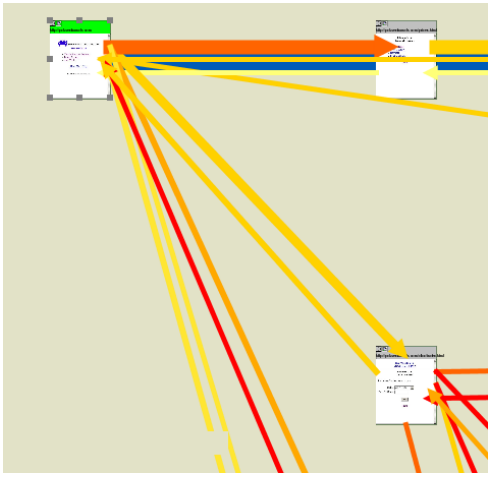

Figure 2: A zoomed in view of the start node in the graph shown in Figure 1. Web pages are now thumbnails and path transitions are more distinguishable.

Figure 3: The start node of Figure 1 zoomed to the page view. Page design is visible, arrows originate from clicked links, and the URL is along the top. Icons on the top are also used to indicate browser back button actions and navigational actions the system cannot associate with locations.

at the thickness of the lines, one can see that many people took the optimal path, but a number of people took longer or different paths.

The semantic zooming and filtering in the WebQuilt visualization are key features for analysis, providing various levels of detail and supporting both the overview of the test and the ability to drill down for specifics. Zooming closer to the paths on the left side of the visualization, highlighted in Figure 4(A), we can see that some users chose to find one piece of information, and then backtrack to the beginning to find the other piece. Designers can use behavior information such as this to recognize places where better navigation may shorten this path, for example by providing more obvious links to remove unnecessary transitions. Two of the ten users following the path highlighted in Figure 4(B) found the correct information, but for the wrong model. This could be interpreted as user error in understanding the task, or a lack of enough visual cues for the user to realize they are not on the correct path. Having more users perform the task or re-wording the instructions may clarify if this is a problem with the site or the test design. Only one of the ten participants exited the task on a page not related to the task.

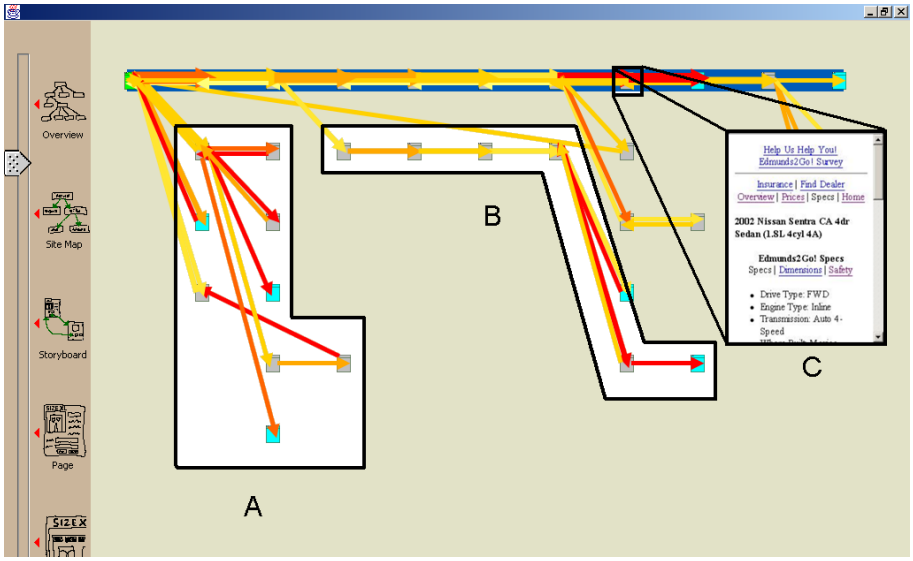

Figure 4: (A) Some users, highlighted in this section of the graph, chose to find the dealer information before the safety information. (B) Most users investigated the most recent Sentra model, the 2002, as the task asked. The two users highlighted here, however, investigated the 2001 model. (C) Users spent a long time looking at this text filled page (superimposed) before selecting the next link.

In this example there are also several red arrows, which indicate that people took a long time – over a minute – before going to the next page. In the overview of the entire task, a viewer can see two thick red arrows along the top. Zooming into these pages, for example the one highlighted in Figure 4(C), we can see that these pages are text-heavy, with long descriptions and lists. It took most of the users considerable time to read through the text and decide where to go next. Because these two pages are in the designers path for this task, and are necessary pages to view in order to successfully complete the task, the designer may want to reconsider how the information is organized on these pages, so as to make prominent the important data or links.

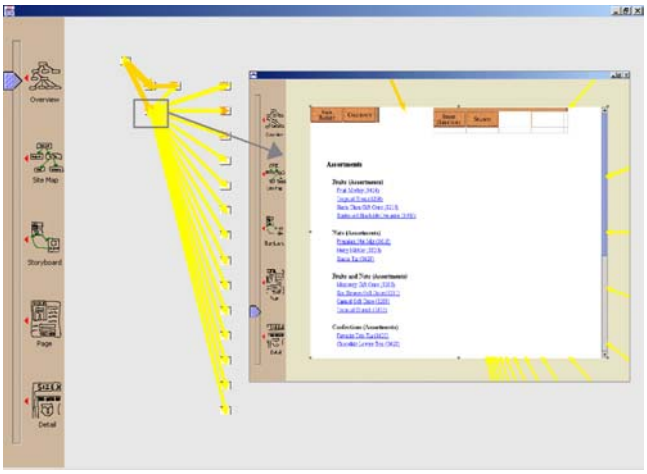

Figure 5: Six users were given the task of finding a piece of information on the Casa de Fruita company web site. The thicker orange arrows at the top show that most of the users found this information in two clicks. One user, however, quickly "ping-ponged" between sitemap and many other pages before finding the information.

Providing the context of the task and a framework to examine more details when needed, this visualization offers a number of simple, but useful analyses of the user experience and behavior. Figure 5 shows the results of a simple information finding task of six users. Most users found the information in two clicks – the top paths in orange. One user, however, "ping-ponged," quickly examining every available link on one page before finding the appropriate information. The WebQuilt visualization allows interesting user behaviors and actions to be identified and analyzed directly with the design of the web pages, so that designers and usability experts can interpret their data within the context of the relevant pages and tasks.

5. SYSTEM IMPLEMENTATION

In the previous section, we demonstrated how potential usability problems and issues can be found using the WebQuilt visualization. In this section we examine the creation of this visualization, from data gathering, to aggregating and inferring user actions, to displaying the interactive graph.

5.1 Gathering Data

At the heart of any data analysis system is the data itself. Gathering quality remote usability data is difficult, as the data is highly dependent upon the logging method used. For a discussion of the advantages and disadvantages of various remote usability clickstream logging techniques, see Hong, et al., 2001 [11]. The

WebQuilt logger is a proxy-based logger. As an intermediary to the client and server, it has the advantage of being able to gather information unobtrusively, without requiring the client or server to have any special software configurations. It can be used to test nearly any web site and is compatible with a range of operating systems, browsers, and devices.

The logging process works as follows. The WebQuilt proxy intercepts each user page requests and relays the request to the content server. When the requested content returns to the proxy, it is modified, redirecting the links to point back through the proxy. A copy of the content is then cached on the proxy machine, and the action is logged in a file format similar to that of server logs, but specialized to include information necessary to recreate the experience for usability analysis. This includes keeping track of the links clicked on each page according to the Document Object Model (DOM), adding transaction ordering identification to pages to recreate the clickstream, and tracking the use of frames. For more information on the implementation of the WebQuilt proxy logger see the descriptions in Hong, et al., 2001 [11]. Additional proxy functionality includes rendering a JPEG image of the web content being sent to the participant, and saving the link locations for reference. Figure 6 shows this process.

5.2 Inferring Actions, Aggregation, Layout

After all of the log files for a task are collected, they are processed to infer where users clicked the back button of the browser. Each log file is transformed into a path of actions, including these inferred actions. It should be noted that, given our logging approach, the system can only be certain that the back button was pressed, but it cannot be certain of combinations of forward and back buttons. The WebQuilt architecture is designed such that developers can create and plug in new models for inferring actions. The processed files are then aggregated into the graph structure, recording the information for each viewing of a web page in a node. This combination of storing information as path traces and as aggregated in nodes allows for the analysis of individual users sessions as well as the overall traffic flow.

The current graph layout algorithm is an edge-weighted, depth-first traversal of the graph, displaying the most trafficked path along the top, and recursively places the less followed paths below it. This algorithm uses grid positioning to help organize and align the distance between the nodes, as well as to avoid overlapping nodes. The graph layout system has been designed to be extremely flexible, and as better algorithms are explored, they can be easily plugged into the system.

5.3 Display

Most of the system has been built on SATIN [12], a Java toolkit for pen-based interfaces. While the WebQuilt visualization is not intended for pen interfaces alone, SATIN provides excellent support for gesture interaction and semantic zooming. To render and save an image of each viewed web page, we use InfoZoom's JacoZoom [14] package to create Java callable wrappers around an ActiveX component containing a Microsoft Internet Explorer browser window. Additionally, using Microsoft's specified DOM components, we are able to record the (x, y) locations of all the links in the web page. These link locations are referenced by the visualization, matching the logged DOM link IDs to the locations on the JPEG, effectively showing which links users selected.

5.4 System Limitations

There are several limitations with the WebQuilt visualization. We cannot infer the resolution of the end user's browser, so our re-rendering is not an entirely accurate recreation of the users' views. The same web page can also appear differently to users based upon the users browser. Our visualization is biased to the rendering of one of the most popular browsers, Internet Explorer. Assuming the participants' browsers and resolutions vary, we are merely selecting one rendition typical for many desktop systems. For smaller devices, such as PDAs, the size of the browser screen used to re-render the content for the WebQuilt visualization can be modified by the designer in order to simulate a variety resolutions.

The WebQuilt visualization system is most limited by the logging technique. For example, the current proxy implementation cannot process all links embedded in JavaScript code. More detailed user interactions, like scrolling and mouse movement, are not captured with this proxy. While this data can be gathered by instrumenting the proxied HTML with data gathering JavaScript, as in Edmonds 2001 [6] or WebRemUSINE [18], it was a conscious design decision to avoid adding any extra information to the proxied content that may limit either the browsers or devices the WebQuilt proxy could log. The current implementation allows designers to explore the world of the Internet beyond the desktop, with PDAs and WAP phones, and still capture and display valuable usability data. However, as the devices vary greatly in the their Internet technology, customizing the proxy for a specific device may allow for logging richer interactions.

6. FUTURE WORK

The WebQuilt visualization system shows promise for helping web site designers and usability experts understand remotely

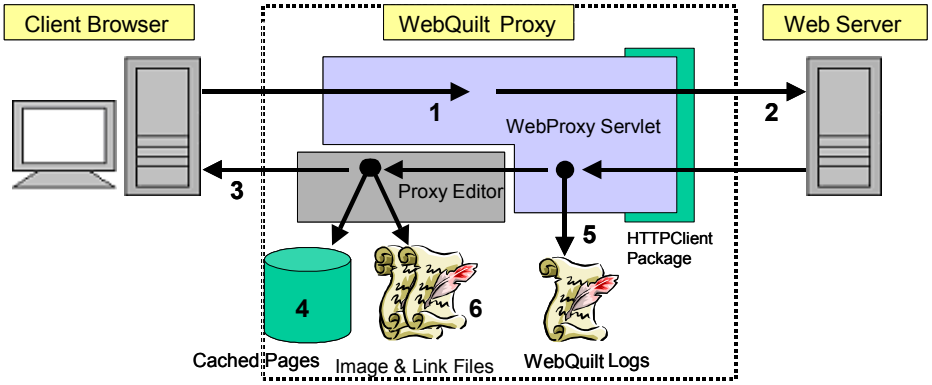

Figure 6: Proxy architecture overview and sequence of operations. 1. The client request is processed. 2. The requested page is retrieved. 3. All of the links are redirected to the proxy, and the web page is returned to the client. 4. The web page is cached. 5. The entire transaction is logged. 6. The JPEG image of the page is created, and the DOM link information saved.

collected usability data [23]. A number of enhancements are currently being added to the system. These include: better navigation support of the graph, customization controls, improved quantitative reporting, and most importantly, improved filtering on a variety of factors and the ability scale to larger data sets. Large data sets require research into clever layout algorithms and heuristics for simplifying clickstream graphs. By combining more advanced filtering techniques with the current visualization system's capacity for semantic zooming, we aim to provide an analysis and interaction framework that is both understandable and usable. Additional plans include an evaluation of the tool with web site designers, as well as incorporating the demographic and questionnaire data into the visualization.

7. CONCLUSION

We have introduced the WebQuilt visualization system for analyzing remote web usability clickstream data gathered by the WebQuilt proxy logger. We have described its implementation and use. Semantic zooming and filtering of clickstream data is shown to be an effective method for exploring and probing usability data, allowing a designer to investigate the data for interesting issues within the context of the relevant web pages and tasks. WebQuilt, the proxy logger and visualization, demonstrate that an unobtrusive usability logging system can provide valuable usability information when used in conjunction with task-based testing and appropriate analysis tools.

Web usability continues to be a very real problem and tools which help designers quickly and easily evaluate web site designs are becoming increasingly important. By enabling designers to visually and interactively evaluate usability data at various granularities, we believe that WebQuilt is a valuable tool for bringing greater usability to the web.

WebQuilt homepage: http://guir.berkeley.edu/projects/webquilt/

8. ACKNOWLEDGMENTS

We would like to send a special thanks to the Group for User Interface Research for their feedback, ideas and support.

9. REFERENCES

1. Brainard, J. and B.Becker. Case Study: E-Commerce Clickstream Visualization. In Proceedings of *IEEE Proceedings of Information Visualization*. San Diego, CA. pp. 151-155, 2001.
2. Carrière, J. and R. Kazman. WebQuery: Searching and Visualizing the Web through Connectivity. In Proceedings of *Sixth International World Wide Web Conference (WWW6)*. Santa Clara, CA, 1997.
3. Chi, E.H., P. Pirolli, and J. Pitkow. The Scent of a Site: A System for Analyzing and Predicting Information Scent, Usage, and Usability of a Web Site. In *Proceedings of ACM CHI 2000 Conference on Human Factors in Computing Systems*. Amsterdam, Netherlands, 2000.
4. Cugini, J. and J. Scholtz. VISVIP: 3D Visualization of Paths through Web Sites. In Proceedings of *International Workshop on Web-Based Information Visualization (WebVis '99)*. Florence, Italy: IEEE Computer Society, 1999.
5. Davison, B. Web Traffic Logs: An Imperfect Resource for Evaluation. In Proceedings of *Ninth Annual Conference of the Internet Society (INET '99)*. San Jose, CA, 1999.
6. Edmonds, A. *Lucidity Project*. 2001. http://sourceforge.net/projects/lucidity/.
7. Eick, S.G. Visualizing Online Activity. *Communications of the ACM*, 2001. 44(8): pp. 45-50.
8. Etgan, M. and J. Cantor. What Does Getting WET (Web Event-Logging Tool) Mean for Web Usability? In Proceedings of *Fifth Human Factors and the Web Conference*. Gaithersburg, MD, 1999.
9. Guzdial, M., P. Santos, A. Badre, S. Hudson, and M. Gray. *Analyzing and Visualizing Log Files: A Computation Science of Usability*. Technical Report, Georgia Institute of Technology, 1994.
10. Hochheiser, H. and B. Shneiderman. *Understanding Patters of User visits to Web Sites: Interactive Starfield Visualizations of WWW Log Data*. Technical Report, University of Maryland, College Park, MD, 1999.
11. Hong, J.I., J. Heer, S. Waterson, and J.A. Landay. WebQuilt: A Proxy-based Approach to Remote Web Usability Testing. *ACM Transactions on Information Systems*, 19(3). 2001.
12. Hong, J.I. and J.A. Landay. SATIN: A Toolkit for Informal Ink-based Applications. In Proceedings of *ACM Symposium on User Interface Software and Technology*. pp. 63-72, 2000.
13. Huang, M. Information Visualization in Web Site Mapping: A Survey. In Proceedings of *World Multiconference on Systemics, Cybernetics, and Informatics*. Orlandlo, FL, July 23-26, 2000.
14. InfoZoom. *JacoZoom*. 2001. http://www.infozoom.de/ie/jacozoom.html.
15. Lamm, S.E. and D.A. Reed. Real-Time Geographic Visualization of World Wide Web Traffic. In Proceedings of *Fifth International World Wide Web Confernce (WWW5)*. Paris, France, 1995.
16. NetRaker. *NetRaker Suite*. 2001. http://netraker.com.
17. Nielsen, J. *Did Poor Usability Kill E-Commerce?* 2001. http://useit.com/alertbox/20010819.html.
18. Paganelli, L. and F. Paternó. Intelligent Analysis of user Interactions with Web Applications. In Proceedings of *IUI: Intelligent User Interfaces*. San Francisco, CA, 2002.
19. Spool, J., T. Scanlon, W. Schroeder, C. Snyder, and T. DeAngelo. *Web Site Usability: A Designer's Guide*. San Diego, CA: Academic Press, 1998.
20. Tedeschi, B. *Good Web Site Design Can Lead to Healthy Sales*. 1999. http://www.nytimes.com/library/tech/99/08/cyber/commerce/30commerce.html.
21. VisualInsights. *eBizinsights*. 2001. http://www.visualinsights.com.
22. Vividence. *Vividence ClickStreams*. 2000. http://www.vividence.com/resources/public/solutions/demo/demo-print.htm.
23. Waterson, S., T. Matthews, and J.A. Landay. In the Lab and Out in the Wild: Remote Web Usability Testing for Mobile Devices. To appear in *Proceedings of ACM CHI 2002 Conference on Human Factors in Computing Systems*. Minneapolis, MN, 2002.
24. Wexelblat, A. and P. Maes. Footprints: History-Rich Tools for Information Foraging. In Proceedings of *Proceedings of ACM CHI 1999 Conference on Human Factors in Computing Systems*. Pittsburgh, PA, 1999.
25. Zona Research. *Shop Until You Drop? A Glimpse into Internet Shopping Success*. 1999. http://www.zonaresearch.com/promotios/samples/zaps/Zap06.htm.