Capturing and Analyzing the Web Experience

Jeffrey Heer
PARC (Palo Alto Research Center)
3333 Coyote Hill Road
Palo Alto, CA 94304
jheer@parc.xerox.com

Abstract
The rapid expansion of the web has made it increasingly important to develop scalable tools for understanding web behavior and identifying usability issues on sites. Research in recent years has explored different methods of capturing and analyzing web usage data for these purposes. In this article we explore two approaches we have developed for capturing web usage data, and highlight two analysis systems that use web usage data to make sense of user activity and determine the goals of web users.

Keywords
Web usability, web logging, web usage, automated analysis, visualization, remote usability testing, web mining

INTRODUCTION

As the Web continues to engrain itself into the fabric of our society, it becomes increasingly important to understand the activities and goals of users and build usable sites that help users achieve these goals. Making sense of the activities of millions of users is an impossible task for an unaided analyst, and so automated means of analyzing web usage data are becoming increasingly popular. Additionally, it can be quite difficult to collect accurate and detailed web usage data in the numbers sufficient to determine significant usage trends and site design problems. As a consequence, remote usability testing is becoming more popular as an affordable and scalable means for evaluating the usability of sites.

This article describes a number of emerging systems that seek to tackle these issues through the automatic capture, representation, and analysis of user behavior. First we explore two different approaches developed for automatically capturing web usage data and examine the tradeoffs involved with each. Next we turn to automated and semi-automated analysis tools for understanding users' surfing actions and inferring the user goals. The goal of such systems is to provide accurate, automated tools that can scale with the growth of the Web, enabling us to understand user behavior and build a more usable web.

CAPTURING WEB USAGE

Over the last few years we have developed systems at both UC Berkeley and PARC for capturing web usage at a higher level of detail than that provided by web server logs. These tools can be used as adjuncts to standard usability testing or to conduct remote usability evaluations. We first introduce WebQuilt, a proxy-based approach to logging user interactions. Next we describe WebLogger, a client-side application which captures a rich array of data.

WebQuilt

WebQuilt is a system we developed at UC Berkeley for conducting remote usability evaluations and logging the resulting clickstreams [6]. What is unique about WebQuilt is its proxy-based approach to web usability logging. It doesn't require users to download and run special software nor does it require web masters to instrument their servers. It can be deployed on any site, or across sites, allowing analysts to run evaluations on any subset of the web that they choose, regardless of who owns the sites in question. The proxy-based approach also allows any client web browser to be used, including those on handheld devices and cellular phones. Furthermore, WebQuilt doesn't function like a standard web proxy that requires configuring the client browser. The end result is that WebQuilt is easy for usability professionals to deploy and easy for participants to use.

WebQuilt works by encoding the necessary information within URLs. When a page is requested by a client, WebQuilt retrieves the page and dynamically rewrites all the links to point back through the proxy before passing the page on to the client. The rewritten URLs contain other useful data, such as an identifier for the current page and individual identifiers for each page link. WebQuilt logs each transaction as it occurs, recording the times, referring pages, links traversed, HTTP methods used, URLs, and any query data (see Figure 1a below for an example of the WebQuilt log format). Additionally, WebQuilt caches each page the user sees, leaving a record of the viewed content. This allows the system to later reconstruct the sessions for analysis purposes, such as that performed by WebQuilt's visualization front-end, discussed later in the paper.

During the implementation of the WebQuilt logger we ran into a number of issues. As the proxy itself must retrieve all the pages a user requests, it must be able to handle all the intricacies of the web. This includes encryption (SSL) support, cookie management, and browser scripting languages (e.g. JavaScript). While we were able to tackle the first two problems, the use of scripting languages can cause a number of complications. It is not uncommon for languages such as JavaScript to dynamically write hyperlinks into the webpage. Since these links aren't created until the page is loaded by the client, the links do not point back to the proxy, creating the opportunity for “runaway” sessions. While a number of heuristics and/or site-dependent hacks can be employed to ameliorate the problem, there does not seem to be a complete, general solution short of including a full script parser into the system. Some scripting languages (or language subsets), however, are browser dependent, further complicating the issue.

Another issue is that analysts may want to capture more data than WebQuilt currently supports. Page scrolling, mouse movements and clicks, window resizes and other low level events are not currently captured. While it may be possible to capture these events by inserting browser-specific scripting code into the proxied pages (see [3]) we have yet to see if this is a truly viable option.

WebLogger and WebLogger-Remote

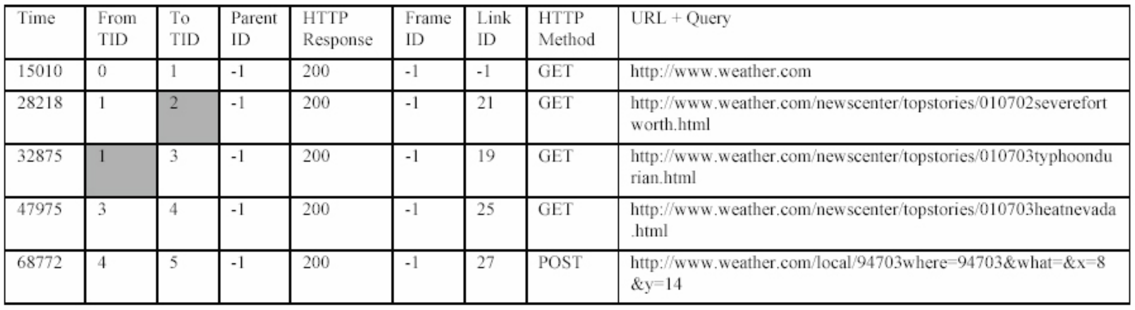
Another tool for capturing web usage data is WebLogger, a client-side browser monitoring system. Developed at PARC, WebLogger was originally intended as both a stand-alone logging tool and as a component of an integrated web eye-tracking system [7]. We have since further developed the system to support remote usability testing.

WebLogger is an application for Microsoft Windows™ that launches an instance of the Internet Explorer browser and monitors and logs user events by using the Component Object Model (COM) and Windows Event hooks. In addition to standard web log information such as viewing times and requested URLs, WebLogger captures a rich set of low-level interaction data such as window resizes, scrolling events, keystrokes, and mouse movements. The WebLogger log format is illustrated in Figure 1b. Pages viewed by users were also cached for later analysis. Since WebLogger uses the Internet Explorer browser, it need not concern itself with many of the intricacies of the web. Secure communication, session management, and scripting languages are all handled by the browser. The caveat is that testing is confined to a single platform and a single browser.

Earlier work at PARC used WebLogger in conjunction with another program named WebEyeMapper. This program maps screen co-ordinates to actual page features such as text segments and interface widgets. It then combines this analysis with eye-tracking data to analyze user surfing patterns. Components of this system were used by Card et al to perform web protocol analysis and automatically generate Web Behavior Graphs [1].

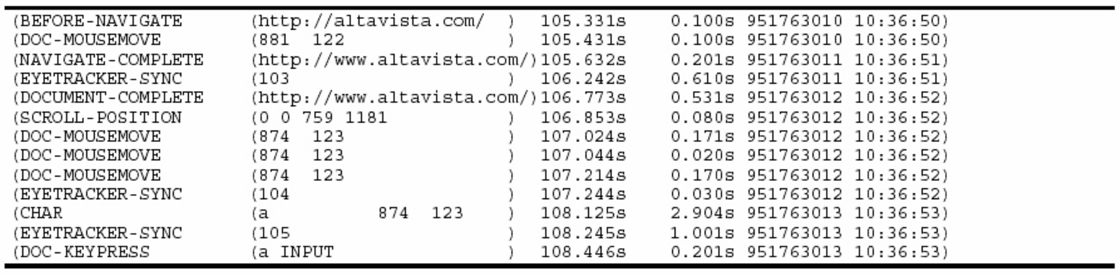
Previously confined to the laboratory, WebLogger has recently been extended to support remote usability evaluations [8]. The augmented system, named WebLogger-Remote, is sent to an evaluation participant who must then run the program on their own machine. WebLogger-Remote then presents them with tasks, and logs the user’s actions. The user log is stored in an encrypted format, preventing the user’s data from being used for unintended purposes.

Once a task has been completed, WebLogger-Remote sends the resulting encrypted log back to a central server and removes the log from the client machine. As a large number of users still connect to the Internet using modems, and we do not want to tie up their bandwidth, WebLogger-Remote doesn’t cache the pages users visit, as sending these pages back to the central server may be too costly. Instead the central server, upon receiving a session log, immediately retrieves the individual URLs viewed and saves its own cached copies. While there are a few cases where this approach may be inaccurate (e.g. if the user visited a site for which they already had some cookie-controlled personalization), we feel it is preferable to inconveniencing test participants.



Time	From TID	To TID	Parent ID	HTTP Response	Frame ID	Link ID	HTTP Method	URL + Query
15010	0	1	-1	200	-1	-1	GET	http://www.weather.com
28218	1	2	-1	200	-1	21	GET	http://www.weather.com/newscenter/topstories/010702severefortworth.html
32875	1	3	-1	200	-1	19	GET	http://www.weather.com/newscenter/topstories/010703typhoondurian.html
47975	3	4	-1	200	-1	25	GET	http://www.weather.com/newscenter/topstories/010703heatnevada.html
68772	4	5	-1	200	-1	27	POST	http://www.weather.com/local/94703where=94703&what=&x=8&y=14

Figure 1a: Sample WebQuilt Log Segment



```
(BEFORE-NAVIGATE      (http://altavista.com/  )  105.331s    0.100s 951763010 10:36:50)
(DOC-MOUSEMOVE        (881  122               )  105.431s    0.100s 951763010 10:36:50)
(NAVIGATE-COMPLETE    (http://www.altavista.com/)105.632s    0.201s 951763011 10:36:51)
(EYETRACKER-SYNC      (103                    )  106.242s    0.610s 951763011 10:36:51)
(DOCUMENT-COMPLETE    (http://www.altavista.com/)106.773s    0.531s 951763012 10:36:52)
(SCROLL-POSITION      (0 0 759 1181           )  106.853s    0.080s 951763012 10:36:52)
(DOC-MOUSEMOVE        (874  123               )  107.024s    0.171s 951763012 10:36:52)
(DOC-MOUSEMOVE        (874  123               )  107.044s    0.020s 951763012 10:36:52)
(DOC-MOUSEMOVE        (874  123               )  107.214s    0.170s 951763012 10:36:52)
(EYETRACKER-SYNC      (104                    )  107.244s    0.030s 951763012 10:36:52)
(CHAR                 (a          874  123    )  108.125s    2.904s 951763013 10:36:53)
(EYETRACKER-SYNC      (105                    )  108.245s    1.001s 951763013 10:36:53)
(DOC-KEYPRESS         (a INPUT                )  108.446s    0.201s 951763013 10:36:53)
```

Figure 1b: Sample WebLogger Log Segment

Although WebLogger-Remote has the drawback that users must download and run the application to participate in a study, it still has a large number of advantages. First, the client-side approach carries no risk of “runaway” sessions; the user can’t inadvertently leave the domain of the logging tool. Second, by employing the Internet Explorer browser, the logging software doesn’t have to focus on re-implementing or working around the messy world of web technologies. Finally, the rich set of data captured allows for analyses other server and proxy-based approaches don’t currently allow. Browsing sessions can be reproduced exactly, capturing the appearance of the browser as the user surfed. This can be useful for discovering important details such as what information on the page was “below-the-fold,” and thus required scrolling to view.

Web Capture Summary

As we’ve seen, there are a number of approaches to capturing user surfing activity, ranging from common server logs to proxy-based and client-side logging. Given the advantages and drawbacks associated with each, the method used should be dictated by the tasks and resources at hand. For example, a proxy-based approach may be faster and easier to deploy when the required information consists only of URLs, viewing times, and links traversed, or if the evaluation spans a number of platforms or devices. However, to validate complex mental models of user activity the rich set of data provided by client-side logging may be necessary.

ANALYZING WEB USAGE

In this section we describe two systems we have developed for analyzing web usage data after it has been captured. The first system is the visualization component of the WebQuilt project, which visualizes user surfing activity recorded from a usability test. The second system is the LumberJack service, which given web usage logs will infer the major groupings of user activity on a site.

The WebQuilt Visualization System

After successfully collecting web user activity data for a usability evaluation, there still remains the daunting task of making sense of the user actions. This is especially true of remote usability evaluations, where much larger numbers of subjects can participate. Since manually sifting through text logs is neither feasible nor enjoyable, we sought to bring the power of information visualization to bear on the problem.

The result is the visualization component of the WebQuilt system, which aggregates the data from evaluation participants and visualizes their activity as a graph [9]. A sample visualization is depicted in Figure 2. The nodes of the graph represent visited web pages and the arrows represent traversed hyperlinks. The nodes are color coded to indicate if a web page is a start page, exit page, or neither. The arrows are color coded to indicate the average time spent on the page before traversing the link. Furthermore, the arrow thickness is used to convey the amount of traffic that went through the link. Designers can also optionally specify an “optimal path” for the current task, which is indicated by thick blue highlighting.

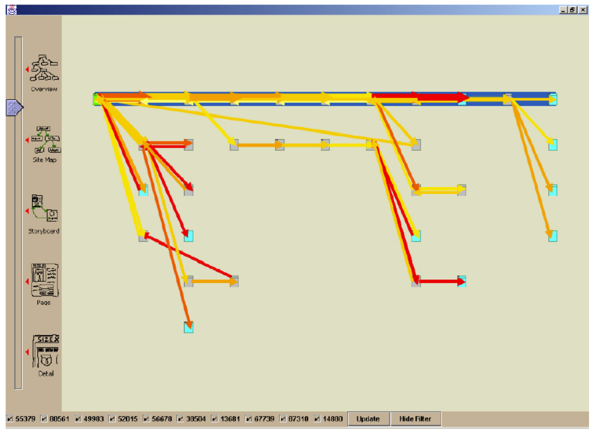


Figure 2: WebQuilt Visualization System

The visualization also supports semantic zooming, allowing analysts to look more closely at a specific area of user activity. When zoomed in, the graph nodes are replaced by actual page thumbnails, and the traversed link arrows will originate from the corresponding link in the thumbnail. Additionally, the interface supports filtering operations that allow an analyst to select which user sessions to display. This could theoretically be used to filter on any number of recorded criteria, including task completion status and user demographics.

Future work for the system might include more in-depth automated analyses of the user paths. Employing heuristics consistent with empirical observation, the system could suggest to the analyst possible “problem areas” where users might have encountered difficulties. Some possible examples include hub-and-spoke surfing (“ping-ponging”), suggesting the user can’t find what she is looking for and/or doesn’t have a specific information goal, and surfing paths that dive into a site and then backtrack heavily, illustrating possible navigational problems. The end goal would be to add more automation to the system to facilitate the process of usability analysis.

The LumberJack Log Analyzer

At PARC we’ve developed an automated web usage analysis tool, which, given records of user interaction with a web site, groups the various extracted user sessions into common activities such as “product browsing” and “job seeking” [4]. We’ve then used this tool to build a new, automated web analysis service named LumberJack.

The system works by first extracting individual user sessions from web logs. Currently the system can use both standard web server logs and WebQuilt proxy logs. The system then crawls the site to be analyzed, and constructs a multi-featured vector space model of the website using the page content, URLs, and hyperlink topology of the site.

User profiles are then created by combining the web site model with the user sessions, using data such as a user's page viewing time to weight the different documents in a user's surfing path. Finally, these profiles are submitted to a clustering algorithm that groups the sessions into (hopefully) meaningful aggregates.

We recently performed a systematic evaluation of different clustering schemes by conducting a user study where we asked users to surf a large corporate site with *a priori* specified tasks [5]. The WebQuilt proxy was used to record user sessions remotely, allowing users to perform the tasks more naturally and on their own time. By knowing what the tasks were and how they should be grouped in advance, we were able to do post-hoc analysis of the effectiveness of different clustering schemes. We discovered that, by counting the number of correct categorizations, certain combinations of data features enabled us to obtain accuracies of up to 99%! Thus by using remote testing tools, we were able to validate our clustering data model of user activity.

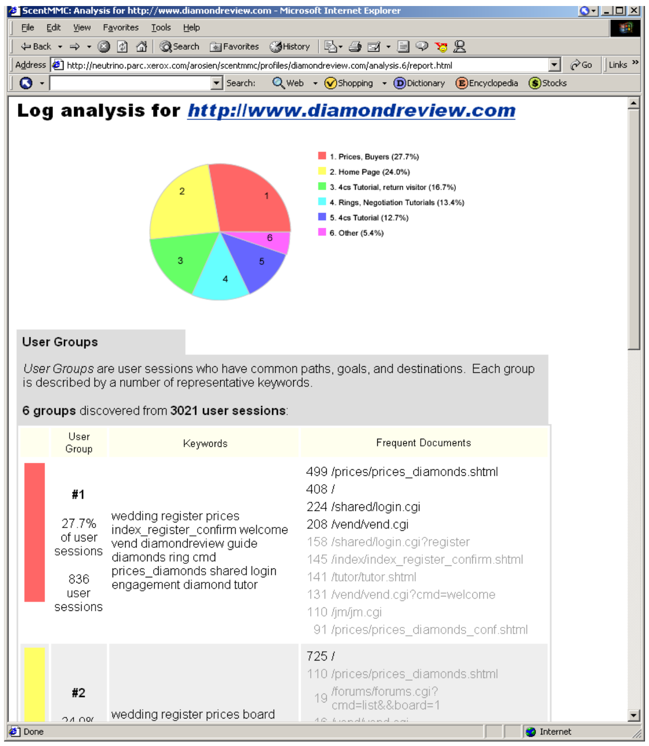
Encouraged by our successful evaluation, we have developed a new service called LumberJack that, given a web server log, automatically crawls the content and hyperlinks of the site and uses this information in conjunction with the user traces obtained from the server logs to discover the user's information needs. This system is a push-button operation, in that only the server logs are needed, and the rest of the system runs completely automatically. Figure 3 shows a screenshot of the report generated for a consumer information Web site on diamonds.

For DiamondReview.com, LumberJack discovered six user groups in 3021 user sessions for 4 days starting from Jan. 4, 2002. 27.7% of the users were buyers interested in prices, while 24.0% spent most of their time on the home page. 16.7% were return visitors going through a diamond tutorial, and we know this because they had obviously started the tutorial from a half-way point that they have bookmarked. 13.4% were ring buyers, and 12.7% were first time tutorial readers. The remaining 5.4% were users that looked at many different pieces of information.

CONCLUSION

In this article we described a collection of web usage logging approaches and subsequent analysis techniques designed to facilitate our understanding of web user behavior. By allowing the automatic capture and analysis of web usage data, such tools carry the promise of affordable, realistic usability analysis and user understanding. The explosive growth of the web and the prohibitive time and resource costs of traditional usability engineering mean such automated tools are not merely attractive, but urgently needed.

Still, there is much progress to be made before true automated usability tests can be a reality. As automated analysis techniques proliferate, analysts will likely want to use a


ScentMMC: Analysis for http://www.diamondreview.com - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://neutrino.parc.xerox.com/arosien/scentmmc/profiles/diamondreview.com/analysis.6/report.html
Go Links
Search: Web Shopping Dictionary Encyclopedia Stocks
1
2
3
4
5
6
Done
Internet

Log analysis for http://www.diamondreview.com

🟥 1. Prices, Buyers (27.7%)
🟨 2. Home Page (24.0%)
🟩 3. 4cs Tutorial, return visitor (16.7%)
🟦 4. Rings, Negotiation Tutorials (13.4%)
🟦 5. 4cs Tutorial (12.7%)
🟪 6. Other (5.4%)

User Groups

User Groups are user sessions who have common paths, goals, and destinations. Each group is described by a number of representative keywords.

6 groups discovered from **3021 user sessions**:

	User Group	Keywords	Frequent Documents
🟥	**#1** 27.7% of user sessions 836 user sessions	wedding register prices index_register_confirm welcome vend diamondreview guide diamonds ring cmd prices_diamonds shared login engagement diamond tutor	499 /prices/prices_diamonds.shtml 408 / 224 /shared/login.cgi 208 /vend/vend.cgi 158 /shared/login.cgi?register 145 /index/index_register_confirm.shtml 141 /tutor/tutor.shtml 131 /vend/vend.cgi?cmd=welcome 110 /jm/jm.cgi 91 /prices/prices_diamonds_conf.shtml
🟨	**#2**	wedding register prices board	725 / 110 /prices/prices_diamonds.shtml 19 /forums/forums.cgi?cmd=list&&board=1

Figure 3: Sample LumberJack report: DiamondReview.com, Jan. 4-8, 2002.

number of different tools and metrics. To this end, it will be quite useful that captured web usage, regardless of how it is gathered, be represented in an accessible format that supports much, if not all, of the information that researchers and analysts will want to apply, and yet is interchangeable across a range of tools. A standard format (perhaps XML based?) for web usability logging could help alleviate such concerns.

Furthermore, to develop tools which can help human analysts pinpoint real web usability problems, it may be necessary to research the existing body of usability engineering knowledge to look for empirically observed indicators of poor usability and/or user confusion (possibly including ping-ponging, backtracking, and excessive viewing times). Equipped with a taxonomy of such indicators, we can then set about the task of determining how such indicators might be algorithmically identified. Alternatively, web usage data collection tools, which allow for meta-data such as task description, task success, and other criteria to be known in addition to raw usage data, can be used to validate independent theories and models of human activity, as we did in [5], and as we plan to do for the Information Scent-based methods for usability analysis described in [2].

While automated tools for understanding user activity and identifying usability problems may never capture all the nuances of standard usability analysis techniques, they offer an avenue for bringing human-centered design and greater usability to a larger segment of the web. It is imperative that we develop these tools to make this possibility a reality.

BIO: Jeffrey Heer is a research scientist at Palo Alto Research Center's User Interface Research Group, with interests spanning Human-Computer Interaction, Computer Systems, and Cognitive Science. His current and previous research projects have explored Remote Usability Testing, Automated Web Usability Analysis, Information Visualization, and Visual Attention. Jeff received his Bachelors of Science in Electrical Engineering and Computer Science (1997-2001) from the University of California, Berkeley. While not actively researching, Jeff enjoys playing the guitar, hiking, reading, and traveling.

ACKNOWLEDGEMENT

The work described in this paper could not have been done without the hard work and collaboration of Ed Chi, Peter Pirolli, Adam Rosien, Rob Reeder, and Christiaan Royer at PARC, and Jason Hong, Sarah Waterson, Tim Sohn, Tara Matthews, and James Landay at UC Berkeley. Work done at PARC was supported in part by Office of Naval Research grant No. N00014-96-C-0097 to Peter Pirolli and Stuart Card.

REFERENCES

[1] Card, S.K., Pirolli, P., Van Der Wege, M., Morrison, J., Reeder, R.W., Schraedley, P., and Boshart, J. (2001). Information Scent as a Driver of Web Behavior Graphs: Results of a Protocol Analysis Method for Web Usability. In *Proc. of the ACM Conference on Human Factors in Computing Systems, CHI 2001*, Seattle, WA.

[2] Chi, E.H., Pirolli, P., Chen, K., and Pitkow, J. (2001). Using information scent to model user information needs and actions on the Web. In *Proc. of the ACM Conference on Human Factors in Computing Systems, CHI 2001*, Seattle, WA.

[3] Edmonds, A. (2001) The Lucidity Project. See http://sourceforge.net/projects/lucidity/

[4] Heer, J. and Chi, E.H. (2001). Identification of Web User Traffic Composition using Multi-Modal Clustering and Information Scent, in *Proc. of the Workshop on Web Mining, SIAM Conference on Data Mining*, Chicago, IL.

[5] Heer, J. and Chi, E.H. (2002) Separating the Swarm: Categorization Methods for User Sessions on the Web. To appear in *Proceedings of the Conference on Human Factors in Computing Systems*.

[6] Hong, J.I., Heer, J., Waterson, S., and Landay, J.A. (2001). WebQuilt: A Proxy-based Approach to Remote Web Usability Testing. To appear in *ACM Transactions on Information Systems*.

[7] Reeder, R.W., Pirolli, P. and Card, S.K. (2001). WebEyeMapper and WebLogger: Tools for Analyzing Eye Tracking Data Collected in Web-use Studies. In *Proc. of the ACM Conference on Human Factors in Computing Systems, CHI 2001*, Seattle, WA.

[8] Royer, C. (2002). WebLogger-Remote. Personal Communication.

[9] Waterson, S., Hong, J.I., Sohn, T., Heer, J., Matthews, T., and Landay, J.A. (2002). What Did They Do? Understanding Clickstreams with the WebQuilt Visualization System. Submitted to *AVI 2002, Conference on Advanced Visual Interfaces*, Trento, Italy.