What is Actually Taking Place on Web Sites: E-Commerce Lessons from Web Server Logs

Mark Rosenstein
Telcordia Technologies, Inc.
445 South St., MCC-1A344R
Morristown, NJ 07960 USA
+1 973 829 4037
mbr@research.telcordia.com

ABSTRACT

A prime business concern is knowing your customer. One legacy carried into the present from the earliest NCSA web servers is web server logs. While there are more powerful user tracking techniques, such as requiring logins or storing cookies, server logs remain a powerful tool in helping understand customer activity on a web site, and are the only tool when logins are not desirable or cookies are blocked by browsers or firewalls. This paper details the possibilities and pitfalls in using web server logs to understand customer behavior on a web site. Described here is the information recorded by the server, and what legitimate inferences can be made from that data. Special emphasis is given to case studies that demonstrate the interactions of the protocols HTTP and HTML, and how weaknesses in the current specification can confound the recorded data and lead to an incorrect analysis.

Keywords

user interaction, web server logs, graphical analysis, visualization, case studies, html, http

1. INTRODUCTION

In the era before the web, obtaining data on customer behavior required either a laboratory experiment with all the artificiality of the laboratory, or explicitly augmenting an interface with monitoring software, with added cost and maintenance nightmares. With the advent of the web came web server logs, which enabled with little additional effort in situ customer behavior monitoring. Just as Prometheus' gift of fire was a double edged sword, so too the analysis of server logs provides opportunities to get burned.

A web server log contains fields that describe each request a browser makes from the server. Through the use of case studies, this paper will describe these fields and reveal the types of ustomer behavior that can be learned from them. Before delving into the precise details of server logs, the first case study will illustrate why a business might care about server logs. This example also strongly hints that a cursory analysis is often insufficient, a theme that will be pursued throughout the paper. All data used here are from actual client e-commerce sites, but for confidentiality the names of the sites will be elided, and only relative traffic levels presented.

A downturn in monthly sales was noted at an e-commerce web site. A quick analysis of the server logs indicated that the site's traffic had also fallen. Looking deeper into the data revealed that the source of most of the site's traffic was from customers clicking on links that pointed to the e-commerce site from the company's main web site. This pattern of behavior was detected using the **referer** (sic) field recorded in the server logs. Careful review of the traffic referred from the main site revealed a radical decrease.

Figure 1 visualizes the drop in customers directed from the main web site utilizing the contents of the **referer** field. The horizontal dimension is time and the height of each bar representing a week's worth of customers referred to the e-commerce site. Further inquiry revealed that the main site had undergone a major redesign, which coincided with the first precipitous drop. We were initially unsure why the traffic continued to fall, but hypothesized that a continuous process of incremental redesign during this period may have contributed to the continuing decline.

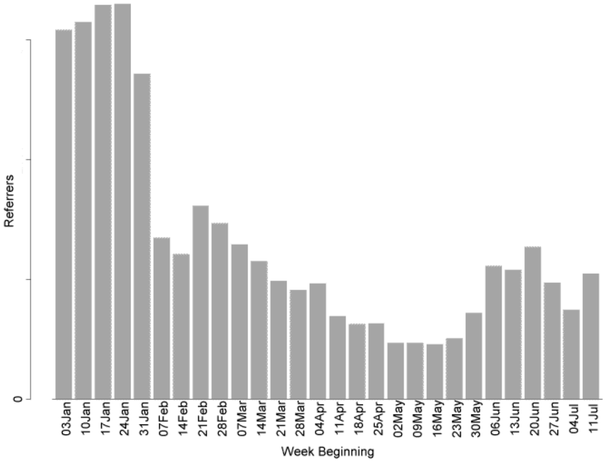

Figure 1 – Customers Referred From Main Web Site

After a flurry of email, and hurried meetings, remedial action was taken on the main site to counteract this apparently disastrous redesign by prominently restoring a link to the e-commerce site on the home page. The upward trend that began in late May was the result of this effort. We will return to this data later, after considering potential pitfalls of this type of analysis. A reexamination will show that this situation was actually not as tragic as the graph portrays. While an actual decrease of referred traffic did occur, the main site's redesign caused underreporting in the **referer** field. This coupled with different reporting behavior from various brands of browsers, coupled with a change in the composition of the browser brands coming to this site all cascaded to cause this graph to be dramatic, but misleading. For instance the referred traffic in July was actually nearly the same as in January.

Judicious use of server logs can provide valuable information on customer's web site actions. In many of the cases that follow, the scenario is an initially plausible but incorrect interpretation, followed by deeper analysis that leads to a better understanding of the situation. This evolution always derives from the fundamental complexity of developing an accurate analysis. Our organization has developed substantial experience in looking at server logs. This paper distills this knowledge in the context of case studies that reveal just how embarrassingly badly things can go wrong and how to derive correct interpretations.

2. WHAT IS CONTAINED IN A SERVER LOG?

Web browsers and servers communicate using the stateless HTTP [1] protocol. The header of an HTTP request message contains attribute-value pairs that a web server can record in its log file, along with information the server can glean from the TCP/IP packet stream. Figure 2 shows typical fields and example values in a web server log entry. Only by combining what can be inferred from this data, coupled with our understanding of how this information was derived, can we accurately analyze customer activity. Each of this section's subsections individually covers in depth a specific field.

```
Originating IP: 198.81.129.99
Timestamp:      [26/Jul/1999:
                10:26:56 -0400]
HTTP Command & Protocol Version:
   "GET /ido/images/id.gif HTTP/1.0"
Status Code:    200
Bytes Transferred: 660
Browser:        "Mozilla/4.51 [en]
                (WinNT; U)"
Referer URI:
   "http://www.company.com/"
```

Figure 2 – Significant Server Log Fields

2.1 The Originating IP Field – Who is Out There

Unless a web site uses an auxiliary mechanism, such as cookies, or requires a login, all that can be determined of a customer's identity from the server log is the customer's IP address. Even if a site tries to use cookies, further identification may be thwarted. Many consumers with privacy concerns block cookies, and in addition many corporate sites block incoming cookies at the firewall, especially companies in security-conscious fields such as finance. This IP address is only tenuously connected to an actual customer. Therefore, extreme care must be exercised in drawing too extensive conclusions from this field.

The second case study demonstrates an incorrect inference. A client came to us with an analysis generated from their server logs by a commercial web log analysis tool. The items available for purchase at this site could broadly be described as "guy stuff," items that would appeal to the male do-it-yourself crowd. In traditional marketing, customer demographics is considered an important component for understanding customers, and one way of deriving customer demographics is based on geographic information. A major reason this client was looking at server logs was to ascertain the geographical distribution of their customer base. Fortuitously, this web analysis tool presented the client's web customers in various geographic ways, including a **Customer by City Report**. An especially puzzling aspect of this report was that a substantial portion of this site's traffic originated from Reston, Virginia.

The client had a couple of theories to explain the large volume of traffic originating from Reston. Since Reston is just outside of Washington, D.C., these theories attributed the traffic to the large retired military population living in Reston, and/or the large number of government employees and government contractors in this area.

Figure 2 clearly demonstrates that there is no geographic information in the server log. The only information available is the IP address of the customer. Any geographic information has to be inferred. It is only possible to make an informed guess as to how this commercial tool derived its geographic information, but it likely took the location registered with the Network Information Center for each IP address as the location of the customer.

When we went back and resolved each IP addresses into originating company name, we immediately noticed that a large Internet Service Provider (ISP) represented a significant portion of this site's traffic. While this provider has dial-up access points located throughout the U.S., their headquarters is located in Reston. We are left to conclude that the geographic inference provided by this tool was less than useful. We believe there are significant questions, given the one-to-one marketing possibilities for web commerce, as to whether geographically derived demographics make sense at all, but certainly incorrect inferences are of no help.

This case study introduces the difficulties in using IP addresses as surrogates for customers. While every computer on the Internet has an IP address, there are a number of ways that the mapping between an IP address and a customer can go wrong. One way is if more than one user has access to the same machine, but at least each temporal sequence of hits belongs to a unique individual. This temporally delimited contiguous sequence of interactions is called a session, so in this case we are able to capture an individual customer's session. More problematic is the presence of intermediary devices between the customer's computer and the web site.

The most common of these devices is a web proxy. The Reston case study can be used to illustrate this problem. When a customer's browser makes a request, the request is routed through a proxy at the ISP. The IP address that is seen at the web site is the IP address of the ISP's proxy, not the customer's machine. Further complicating the matter, the ISP has a number of proxies, and each request the customer makes may go through a different proxy. As there are many users of the ISP proxies at any one time, it is both possible that a single user's visit will appear as multiple IP addresses (going through different proxies), and different customers will appear with the same IP address.

To measure the magnitude of this problem, we instrumented a business-to-business site to issue a session id when a customer first retrieved the site's home page. All links from the home page contained this session id in their URLs, so even if a customer made requests via multiple proxies we could keep track of the user, at least for the length of the session. At this site between 10%-20% of the traffic consisted of multiple IPs for a single customer.

2.2 The HTTP Command Field – What Was Requested

The **HTTP Command** field contains the browser's request for server resources. One of the simplest and most frequent requests is the **GET** command, which causes a page to be retrieved. The command field is useful for determining customer's activity on a web site. There are two major types of analysis possible with this information, an analysis on a per-page basis, and a session analysis.

Per-page analysis requires fewer assumptions and can produce usage characteristics, such as listing the popular pages on a site. Session information provides the more interesting commentary on a site, since it gives a better feel for the interactive nature of the customer's experience. As can be seen from the difficulty in resolving IP addresses to unique customers, constructing sessions from log data is problematic.

As per-page analysis is often well supported by commercially available tools, only a short case study is necessary to understand its use. For a site rich in content, designers carefully researched the trade-off between page length and complexity of links. As a result, they chose a site design that initially showed customers a main page consisting of a table of contents requiring almost no scrolling, but displaying very little actual site content. The results as reflected in the server log files were unpleasant. Over 80% of the customers left the site after viewing only the table of contents.

A redesign was initiated and the site flattened to have a substantially longer home page, now augmented with a sampling of the site's content. Reviewing Figure 2, we see that server logs do not directly indicate if customers actually scrolled down a page to view its content. However, in the redesigned site, content links far enough down the main page to requiring scrolling at most screen resolutions were clicked on more frequently than in the previous scroll-free design. This provides indirect evidence that the new design was working. After a few months' experience with this design, the number of customers leaving the site after viewing only the first page had dropped to approximately 50%, a significant improvement.

This result stands independent of a designer's theoretical underpinnings or taste in guidelines. Server logs provide concrete evidence on whether these models of user behavior apply for a specific site. We believe the improved performance of the redesign results from a combination of increased "scent" [4] on the first page, as well as enticing customers to stay on the site, for the same reason that the display windows of department stores contain merchandise and not just posters with a list of the categories of items sold in the store.

Even in a per-page analysis there are some mysteries that can not be fully resolved by web server logs. In designing a web site, there is a constant tension between aesthetics and download time. Often a fancier presentation represents more bytes, which takes longer to download from the server. On one page of an active site, there are three images of respectively 394, 524, and 16478 bytes in size. A priori, one would expect these images to be downloaded with approximately the same frequency. While the server logs show that the first two images received almost exactly the same number of requests (less than .1% difference in hits in a typical week) the third image consistently only received about 2/3 of the hits of either the first two.

We have two hypotheses to explain this anomaly. The first is that customers are hitting the **STOP** button on their browser which interrupts the transfer of a page. In a typical browser, as the downloading page is parsed, the browser begins to download the embedded images. This theory's claim is that the two smaller images complete their download, and the user runs out of patience waiting for the large image and presses the browser's **STOP** button, aborting its transfer. The second theory is that the customer finds a relevant link while the large image is still downloading and clicks on it. This also causes the browser to terminate the download and begin to load the page referenced by the link. Neither claim can be verified, since this information is not recorded in the server log. Only by recording the underlying TCP/IP stream can this hypothesis be tested. This data does bring into question the overall value of the large image, which 1/3 of the customers seem willing to forego.

As mentioned above, analyzing the complete user session is fundamentally more interesting, since it details the entire user experience. Unfortunately, there are additional problems in reconstructing accurate and complete sessions, even if auxiliary session information (a session id) is recorded in URLs, hidden variables or cookies. The server log can only record requests that actually appear at the server. The most common reason for a customer's action to not generate a server request is caching. Most browsers, by default, cache pages they receive. This means that a local copy of the page is kept on disk until some expiration time is exceeded. When the customer revisits a page, the browser first checks its cache, and if the page is in the cache, the page is served from the local disk, and no request is seen at the server. This implies that many circular paths through a site are poorly represented in server logs.

Many sites depend upon the use of the browser's **BACK** button for navigation. The **BACK** button allows a customer to return to a previously visited page (usually the immediate predecessor to the current page). With caching, these navigation actions are invisible to the server log, though they may predominate in the customer's behavior.

The second mechanism in which requests do not appear in server logs are caching proxy servers. Some sites from which customers originate implement a cache between their users and external web servers. In this design, the first request for a page from a user behind the cache is transmitted to the server, returned to the user and also stored locally on the proxy. All additional customer requests for that page from behind the proxy will receive the page from the proxy instead of from the originating server.

These factors make us very cautious in attempting to build meaning out of sessions constructed from web server logs. These difficulties may also effect quantitative measures of web site effectiveness[3]. A heavier weight mechanism than log files seems necessary, and the Future Work section indicates one technology that we have developed.

2.3 The Browser Field – What is Out There

The **browser** or **user-agent** field indicates which browser a customer is using to access the site. This field is supplied by the browser and can be used to understand the mix of browsers accessing the site. Also, well-behaved webwalkers identify themselves in this field. Many sites choose not to record this information, because they believe it to be irrelevant, which is unfortunate, as the next case study shows.

As part of an analysis for a client, we looked at summary statistics of their e-commerce site's traffic. A site with predominately U.S. customers would have a strong 9am to 5pm traffic bias, with longer tails caused by activity spread over the three U.S. time zones. This site showed a fairly uniform distribution of visits measured by both time of day and day of week. Figure 3 is a time of day graph with the height of the grey bars indicating the hourly level of traffic for a typical week in February 1999. Note the low periods of traffic differ from the peak traffic by only about a factor of two, which differs significantly from the expected U.S. only pattern.

The client inferred from this result that the substantial, off-peak traffic indicated significant visits from outside the United States. The site's owners were quite pleased by this result, because attracting foreign customers was a priority. There were two nagging questions about the traffic, though. The first was a very low buy to browse ratio. The second was that none of the highest volume customers were familiar, existing corporate customers. Customers were determined by resolving the IP addresses in the server log. This second factor was not investigated as thoroughly as it should have been, due to the large number of new entrants in this client's business, and the belief that the availability of on-line purchase was attracting customers from previously unexploited market segments.

On further investigation, approximately 60 of the top visiting IP addresses all had a surprisingly equivalent number of requests. Analyzing each of these visits revealed that only the front page of our client's site was being requested, and the request was being repeated once each hour.

The breakthrough came when they turned on recording for the **browser** field. For these 60 hosts, the browser field contained the name of a web monitoring tool. What was generating this unexplained traffic was a program, not paying customers! A factor that delayed this discovery was that this traffic was spread across multiple IP addresses, and across different companies in different locations. A phone call to the monitoring company revealed that the web site's own IT organization had contracted for this "service". However, the group responsible for the content of the site and its traffic analysis had never been informed. Once these probes were filtered out, our client's view of what was actually happening on their web site significantly changed. The black bars in Figure 3 show traffic with these monitoring sites filtered out. With the monitoring traffic removed, the site exhibits a normal business hour traffic pattern.

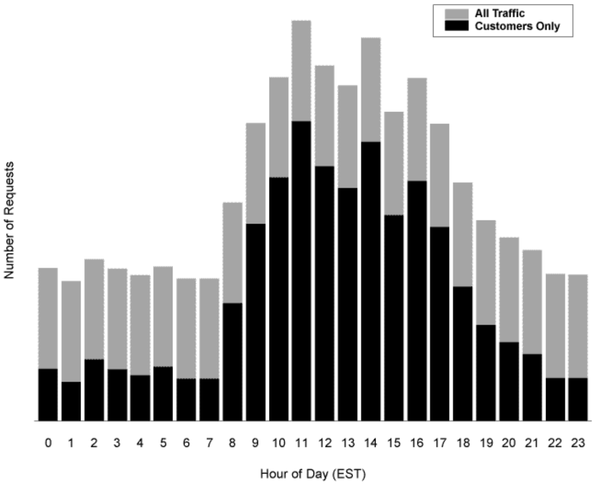

Figure 3 – Summary Hourly Traffic

2.4 The Referer Field – How Did They Get Here

Web browsers provide a field in the HTTP header called **referer** which holds the page containing the link that a customer clicked to generate this request. This field is useful to discover how customers get to a given page. The analysis of this field is problematic, and this subsection returns to the data from the first case study for the most complex analysis in this paper. The extended minuet between the data and the analyst that was necessary to deeply understand the behavior encompassed by the first case study follows.

The HTTP specification is notably vague on the specifics of how this field should work. "The Referer [sic] request-header field allows the client to specify, for the server's benefit, the address (URI) of the resource from which the Request-URI was obtained..."[1] It is clear what an analyst requires from this field: the page that customers are clicking on to be directed to their site. Unfortunately, the technology that generates, stores, and reports referrer information does not, in many cases work as expected. To see how this process can fail, it is necessary to first understand how referrer information is generated.

When a customer clicks on a link, the HTTP request sent to the server contains in the **referer** field the URL of the page containing the link. The analyst is at the mercy of the web browser to provide this information correctly and consistently. We only consider the top two brands of browsers, since for this study they account for nearly all of the customer traffic. Where it is necessary to distinguish between these two brands, they will be referred to as Browser1 and Browser2.

The first HTTP mechanism where ambiguity might cause inaccurate reporting is **redirects** or server forwards. This common mechanism is invoked when a browser requests a page and the server response (response-code 302) instructs the browser to fetch a different page. The browser must then decide what (if any) referrer to send with the request for the new page specified in the redirect message. The top two browsers both pass the original referrer for this subsequent request. Note that for the servers we examined, one customer click generates two log entries, one for the original request and one for the redirected request. To get an accurate count of actual customer driven refers, just counting referrer entries in the log file is insufficient, but instead the entries with status code 302 must be ignored. This adjustment was correctly made in the data reported in Figure 1.

A second place where discrepancies might arise is from web pages that contain within them HTML[5] tags that generate requests for additional entities (such as pages) from the server. The simplest case is a page with a reference to an image. Define the site of interest as the site we are analyzing. Define an external site as the site where a customer generates a page request to the site of interest by clicking a link on the external site. The goal is to accurately count customers arriving via the external site’s link. Call E a page on the external site with a link to H a page on the site of interest. Call I an image referred to on page H. When the customer clicks on the link on E to go to page H, the browser sends a request for page H and passes on E as the referrer. As the browser receives and parses page H it recognizes an HTML image tag on page H and now must retrieve image I. What should the request for image I pass as the referrer, E or H? Unfortunately, this is not spelled out in the HTTP protocol, but luckily, the two most popular browser variants agree it is H, which seems correct, since H specifies the request for I.

A similar construct using frames has a more complex behavior. The home page in the first case study consists of a frame composed of two panes. In HTML the home page contains a **FRAMESET** markup, which specifies the URLs and layout of the two panes. Again, E is the external page with a link to H the homepage, and F1 and F2 are the two panes composing the frameset described by H. When a customer clicks on the link to H on page E, the browser requests page H and sends along E as the referrer. After parsing H, the browser realizes it needs to request F1 and F2. Different brands of browsers, and even different versions within a single brand, have different behavior in reporting a referrer for F1 and F2. To be consistent with the embedded images mechanism, H should be passed along as the referrer for F1 and F2. Browser1 passes E as the referrer for F1 and F2 as well as the referrer for the frameset page H. The latest version of Browser2 passes E as the referrer for page H, but passes along H as the referrer for panes F1 and F2.

While this divergence is not ideal, it should be possible to use the **browser** field to distinguish these two cases and correctly account for referrers. Unfortunately, previous versions of Browser2 are inconsistent in reporting referrers in the frame case. It appears to the author that in some minor release of this browser, the behavior changed, but unfortunately, the minor release numbers are not reported in the **browser** field, so that in the general case there is no simple, automatic procedure to compensate for this inconsistency. In this specific case, though, it is possible to derive consistent results. For this site, customers can only enter the site via page H, so all referrers to panes F1 and F2 can be discounted.

Notice that if the mix of browsers changes over time, even with the same level of actual referrer activity, the reported activity will change. This is exactly what is happening during the period depicted in Figure 1. The mix of browsers was shifting towards the latest version of Browser2 and the market share of this browser was increasing relative to Browser1. Given that Browser2 only reports one referrer for each actual referrer, while earlier versions of Browser2 and all versions of Browser1 report three referrers, this change accounts for much of the linear decline in Figure 1.

We now come to the mechanism that caused a major portion of the dramatic drop in Figure 1. The redesign of the main site specified that links to the e-commerce site went through a jumping-off page on the main site. This page had a link to the e-commerce site. It was decided after the initial deployment that this extra step was unnecessary, so the content of the intermediate page was replaced by the following html:

```
<HTML>
<HEAD>
<META HTTP-EQUIV=”REFRESH” CONTENT=”0;
URL=http://ecommerce.company.com”>
</HEAD>
</HTML>
```

When this markup is parsed by the browser, it tells the browser that after 0 seconds (i.e. immediately) retrieve and replace the current page with http://ecommerce.company.com. This is called a **Browser Pull** page.

So if site F has a page B, containing the above markup, and page E, also on site F has a link which points to page B, when a customer clicks on the link on page E, the browser retrieves page B (which remember is on site F, not the e-commerce site) and an entry in site F’s server log is made with referrer E for this page B. Then zero seconds later, the browser follows the meta markup on page B and requests the home page from the e-commerce site. What referrer should the browser pass along? In the case of the two leading brands of browsers, nothing is passed as the referrer, and the referrer is recorded in the log file as “-”. We have found one exception to this rule in which Browser1 on the Apple Macintosh(tm) platform reports page B as the referrer. This is almost certainly the correct action, because it follows the same model as images. Clearly, the design change of the main site caused a huge underreporting of referred traffic witnessed in Figure 1.

This little dance was not at all pretty, but indicates what is possible and necessary to correctly interpret log data. This issue was critical for understanding what was actually happening in the first case study. In our final analysis, it turned out that referrers had actually fallen approximately 30% due to the new design, not the factor of 3 indicated in Figure 1. A redesign was still called for, and was executed, which brought traffic back up to nearly its former level.

3. FUTURE WORK

Given the difficulties outlined above, an area that calls out for further inquiry are other classes of web site statistics. One of the trickiest measurement problems is the effectiveness of advertisements, especially banner ads. The difficulty of measuring this effectiveness has led some [6] to look for alternative technologies to banners.

We have completed some very preliminary work in this area. Working for a client, we looked at the results from one site on which they had purchased banner space. From that site for June, 1999, the vendor provided the following none too impressive data:

Impressions	14,000
click-through	105
%	0.75

Here impressions are the number of times the banner was shown, and click-throughs are the number of "customers" that clicked on the banner. Making this data amenable to further analysis, the vendor also provided the times and the host names for each click-through. A web site analysis tool was run against the server log data. We found using the **referer** field that only 57 click-throughs had been referred to the client's site from the banner ad site.

A quick analysis showed the 57 hits were what our client had hoped for: sessions with human-like characteristics, driving real, potential buyers to the site. This inference was based on the customers actually receiving the pages and having a **user-agent** (browser) that people normally use. The task remained to explain the other 48 click-throughs.

Almost all the rest of these click-throughs turned out to be web walkers. A broader search of the log files revealed that many of the unaccounted for hits had specific web walker identification in the **user-agent** field, confirming our suspicions that the missing traffic was not driven by customers. It is possible that a few of the IPs outside of the 57 were actually customers, but at least at a first pass only 54% of the click-throughs were customers, who actually viewed the client's home page. The rest were likely to be just internet noise.

Besides expanding this work into other areas, we hope this work encourages the strengthening of existing standards, to guarantee that data is gathered and recorded in a useful, consistent manner. We have also built some monitoring equipment that will help us in building understandings from the TCP/IP packet level up to the user action level.

Finally, we have been looking at heavier weight solutions to the problems detailed above. If one were to engineer monitoring of web page use, it is unlikely web server logs would be the solution. An obvious omission is that there is no way to tell when a customer has left a page unless the customer goes to another page on the site, generating a visible log entry.

Given our interest in recommendations[2], we have become very interested in using filtered time-on-page as a surrogate for interest. To receive time-on-page data regardless of caching, and the use of the **BACK** button, we felt we had to move to a more intensive technology.

In the Personal Site Navigation system, we prototyped a very small Java(tm) applet that would be placed in one or more locations on each web page on a site. This applet reports back to the server using extensions to the HTTP protocol with timestamped records of a customer's entry and exit from a page. Just as with cookies, users can protect their privacy by opting out of this facility.

We have experimented with various versions of the applet, some of which allowed explicit ratings as well as implicit time-on-page. Despite the added overhead, we believe that for some applications the added functionality provided by this type of technology nets a positive return for the customer.

4. ACKNOWLEDGMENTS

The author would like to thank his project team, especially Janet Salvi and Jack Shea for the opportunity to work with interesting data, and also the Datamining "skunk works" group headed by Dennis Egan, who provided help and comments as this research was being conducted. Credit goes to Marc Fusco for the Reston example. The author would also like to thank Carol Lochbaum, Eric Sigman, Dannie Durand and Louis Weitzman, who reviewed early drafts of this work and generated insightful comments.

5. REFERENCES

[1] Fielding, S., et. al. RFC2616 Hypertext Transfer Protocol -- HTTP/1.1. Available at ftp://ftp.isi.edu/in-notes/rfc2616.txt. June 1999.

[2] Hill, Will, et. al. Recommending And Evaluating Choices In A Virtual Community Of Use, in Proceedings of CHI'95 (Denver CO, May 1995), ACM Press.

[3] Lohse, Gerald L., and Spiller, Peter. Quantifying the Effect of User Interface Design Features on Cyberstore Traffic and Sales, in Proceedings of CHI '98 (Los Angeles, CA, April 1998) 211-218.

[4] Pirolli, P. and Card, S. Information Foraging in information access environments, in Proceedings of CHI'95 (Denver CO, May 1995), ACM Press, 51-58.

[5] Raggett, Dave, Le Hors, Aranud, and Jacobs, Ian (eds.) HTML 4.0 Specification. Available at http://www.w3.org/TR/REC-html40/. April 1998.

[6] Risden, Kirsten, et. al. Interactive Advertising: Patterns of Use and Effectiveness in Proceedings of CHI'98 (Los Angeles, CA, April 1998) 219-224.