Evaluation of Session Identification Heuristics in Web Usage Mining

Amithalal Caldera and Yogesh Deshpande
School of Computing and Information Technology
College of Science, Technology and Engineering
University of Western Sydney
PO Box 1797, Penrith South DC, NSW 1797, Australia

h.caldera@uws.edu.au, y.deshpande@uws.edu.au

Abstract

Web Usage Mining (WUM) is the discovery of interesting knowledge from Web server logs. The access log files of a Web server contain a lot of details about users' on-site behaviour. The validity of WUM depends on the accurate identification of user sessions implicitly recorded in these logs. In some applications, a user may be explicitly identified through user authentication. However, in general, the Web logs do not contain a user id and separate user sessions have to be inferred through heuristics. This is generally difficult because of several additional factors, such as Web caching, the existence of proxy servers and the stateless service model of the HTTP protocol. Several heuristics exist to address these problems. By definition, the heuristics yield inexact and variable results. It is, therefore, crucial to analyse and understand how good a particular heuristic is likely to be in a given environment. This paper reports on an investigation into the performance of a composite heuristic based on three published heuristics found in literature to identify sessions from the Web logs. We use the logs of a university Web server that records user ids for administrative reasons, which allows us to evaluate the heuristics against the concrete knowledge of user sessions. Consequently, the paper also proposes a strategy for future log analyses and makes recommendations for further work.

Keywords: heuristics, user tracking, user identification, session identification, log analysis, web usage mining.

1. Introduction

Web Usage Mining (WUM), a discovery of interesting user access patterns from Web server logs, has become the subject of intensive research, because of its potential for personalized services, adaptive Web sites, target marketing in e-commerce, and organization and presentation of Web sites. With the transformation of the Web into the primary tool for electronic commerce, it is imperative for organizations and companies, who have invested millions in Internet and Intranet technologies, to track and analyse user access patterns hidden in their Web server logs.

A Web server log is an important source for performing WUM because it explicitly records the browsing behaviour of site visitors. It provides details about file requests to a Web server and the server response to those requests. However, because of considerations of privacy, the logs, by default, do not record user ids.

For meaningful WUM, on the other hand, these requests must be identified into user sessions as semantic units of analysis. The difficulty of identifying users and user sessions from Web server logs has been addressed in by several researchers [1, 2, 3]. A solution to this problem is to create heuristics that capture in a logical way the behaviour of users and map it onto the Web logs.

The researchers have also examined the performance of the proposed heuristics[3, 4, 5]. This paper is a contribution to this analysis, based on the logs of a University Web site. These server logs, for operational reasons, maintain explicit user ids. The server under consideration also serves a fully known set of users, viz. students. The conditions under which the students work are also known. We can, therefore, analyse the logs to know exactly which student worked on the server at what time and for how long. Hence, by correlating this analysis with that thrown up by heuristics we are able to arrive at how well they perform. While these results are site-specific and hence difficult to generalise, the analysis enables us to formulate and propose a strategy for future analyses under different conditions and improve one’s understanding of user behaviour of a particular site.

The rest of this paper is organized as follows. Section 2 describes the logged data in general and the difficulties in using such data to analyse the user behaviour. Section 3 is a brief survey of the available heuristics and related work. Section 4 describes the heuristics combined for the evaluation. Section 5 explains the methodology used for the evaluation of heuristics. In section 6, we compare the performance of the heuristics for the site under consideration. Section 7 proposes a strategy for further work in evaluating heuristics and concludes the paper.

2. Web Server Logs

Web server log files are the primary source of data in which the activities of Web users are captured. These log files can be stored in various formats such as Common Log Format (CLF) or Extended Common Log Format (ECLF) as recommended by W3C[6], Microsoft and NCSA. An ECLF file is a variant of the CLF file simply adding two additional fields to the end of the line, the referrer and user agent fields. Each entry in the log file stored in ECLF describes the source of a request, the file requested, the date and time of the request, the URL referring to the requested file (referrer), the client environment (user agent), and other data such as server return code and the number of bytes transferred.

User requests for one URL frequently result in multiple entries in the server logs, independent of one another, representing requests to the server for each of the hyperlinked elements, such as images, style sheets and so on. The number of requests per day of a medium-large Web server is in the order of millions and a popular Web site can see its Web log growing by hundreds of megabytes every day.

The stateless service model of the HTTP protocol does not allow support for establishing long-term connections between the Web server and the user. The lack of explicit user identification in the log means that even the multiple requests generated by a single click cannot be assigned to the individual user who has initiated it with 100% certainty.

The process of user identification is mostly based on the IP address of the client machine that made the requests. This IP address may be of a machine used by only one user at a given time or it may be that of a proxy in which case it could represent a number of users whose requests are being routed through it. In the first case, the same machine may be used by different users over time, a fact that is impossible to deduce from the log data. In the case of a proxy, numerous requests to the Web server from users connected to the proxy can occur simultaneously. In both cases, the tasks of inferring from the log data the individual users and the 'paths', i.e. the hyperlinks, traversed by each of them become non-trivial. There is another impediment to this process of user identification, viz. caching. There are various levels of caching embedded in the Web, mainly to expedite a user’s access to the frequently used pages. Those pages requested by hitting the “back” button available in most browsers, (heavily used by the Web users [7]), are all retrieved from the Web browser cache. Also, proxy servers provide an intermediate level of caching at the enterprise level. The server log data cannot capture these cache hits, rendering it an incomplete source of user behaviour.

3. Related Work

Researchers have proposed various methods to resolve the problem of tracking users and their activities from the server log data and also highlighted their drawbacks.

Client-side data collection can be implemented by modifying the source code of an existing browser to enhance its data collection capabilities. A modified browser is much more versatile and will allow data collection about a single user over multiple Web sites. In [8], XMosaic 2.6 was modified to record a user’s browsing activity. The most difficult part of using this method is convincing the users to use the browser for their daily browsing activities.

Client-side data collection can also be implemented by using a remote agent. A remote agent developed as Java applet was introduced in[9, 10] as a client-side Web usage data acquisition system. When a user first enters a Web site, the remote agent is uploaded into the browser at the client side. Thereafter, it captures all required features of user interactions with the Web site and transfers the acquired data to a data acquisition server, called the acquisitor. When the agent is uploaded to browser, it receives a globally unique session ID from the acquisitor and labels all captured data sent to the server with that ID. In addition, the remote agent reports visiting of cached pages (whether at the proxy or at the browser) to the acquisitor server which results in more accurate tracking as compared with what records at Web server logs. Thus, the aquisitor can transparently store data captured by different agents as separate semantic units, i.e., user sessions, without further requirement for user session re-identification. The main drawback with this implementation is that running the remote agent at the client side requires users cooperation in enabling Java at their browsers.

The data mining results from the 1996 Olympics site [11] were obtained using cookies to identify site users and user sessions. An HTTP cookie issued by the server to the client browser identified a visitor to the Olympic Site. Since the server logged the cookie of every request, the requests coming from any given browser could be positively identified. The major downside of this method is that many users often choose to disable the browser features that enable the acceptance of cookies. It is also impossible to know whether more than one person visits the Web site using the same instance of a browser.

An innovative method, called page conversion was introduced in [12]. This mechanism involves software downloading and works as follows. First, a Web page is encoded into cipher by a server-side enciphering module. The original Web page is replaced by this enciphered Web page. Then, a client-side program, called deciphering module, deciphers these encoded data and displays the content to Web user. The deciphering module also reports the user behaviour to the Access Pattern Collection Server (APCS) before the data is deciphered and shown. By having the enciphering and deciphering mechanism, one can ensure that these Web pages will not be shown unless the deciphering module is called and the APCS is informed, preventing deliberate bypassing of the data collection process. (Of course, the enciphering/deciphering mechanism can be removed if the system allows users to bypass the data collection process.). Each line of the APCS log consists of access time, user name, host name, host address and the URL of the Web page accessed. With user access patterns properly collected, the individual Web user behaviour can be better captured and analysed by the corresponding data mining techniques. The violation of user privacy is the main concern in this method.

Lamprey [13], a tool for doing quantitative and qualitative analysis of Web-based user interfaces, tracks users by rerouting all of their Web navigation through a central tracking gateway. The central mechanism of Lamprey's user tracking system is the parsing of HTML pages and embedding of tracking information in every hypertext link in the page. When a user being tracked by Lamprey requests a page, the system fetches it and changes every URL in that page to reroute it through Lamprey. An altered URL includes all the parameters necessary for Lamprey to fetch the original page and return it to the user. Once the user sends a URL to the Lamprey application, all subsequent activities are logged in Lamprey log files.

A technique of dynamic page rewriting is used in [14, 15, 16] to track the user. In this method, when the user first submits a request to the Web site, the server returns the requested page rewritten to include a hidden field with a session-specific ID. Each subsequent request of the user to the server, will supply this ID to the server, thus enabling the server to maintain the user's navigation history. An identifier timeout mechanism was also used to make sure different sessions from the same client are given different identifiers. This session-tracking method does not require any information on the client side and can therefore be always employed, independently of any user-defined browser settings. But this method restricts intermediate caching and does not correctly handle the exchange of URLs between people.

The heuristics proposed in [1, 3] can be used to help identify user sessions with relative accuracy in the absence of additional information such as cookies, user id or session id. The first heuristic states that two accesses from the same host but with different browser versions or operating systems are initiated from different visitors. The second heuristic states that if a web page is requested and this page is not reachable from previously visited pages, then the request should be attributed to a different user. The third heuristic is that all requests from the same host, browser and operating system within a threshold (usually 30 minutes) are considered to be part of the same session. The fourth heuristic says that the time spent on a page must not exceed a threshold (usually 15 minutes) to be part of the same session.

4. Selection of Session Identification Heuristics for Evaluation

In this study, we evaluate the performance of a composite heuristics based on three published heuristics. These heuristics are mainly to overcome the problems of proxies as well as to identify multiple users with the same, genuine IP address, as, for example, happens in a lab. The description of the three heuristics is as follows.

IP/Agent: Each different user-agent type for an IP address represents a different user.

A user-agent is an arbitrary assigned string, which shows a change in use of browser or operating system. The rationale behind this rule is that a user rarely employs more than one browser when navigating in the Web.

Referrer: A referrer is the URL of the page the client was on before requesting the current page. If a page is requested that is not directly reachable by a hyperlink from any of the pages visited by the user, then the request should be attributed to a different user even if the IP/Agent string is the same for the two consecutive page requests in the log.

The rationale behind this heuristic is that users generally follow links to reach a page and very rarely type URLs and use bookmarks.

Timeout: For the same combination of IP/Agent, if the time between two consecutive page requests exceeds a certain limit (15 minutes), it is assumed that the user is starting a new session.

The motivation behind this heuristic is that for logs that span long periods of time, it is very likely that users will visit the Web site more than once. Visitors who do not request pages within a certain time limit are assumed to have left the site and started new session.

The third and the fourth heuristics mentioned in the last paragraph of section 3 concerns about the timeout of a session and a page respectively. We combine the fourth heuristic with the first two in our evaluation

5. Methodology for Evaluating Heuristics

5.1 Environment

The Web logs used in this investigation came from the server for a student lab used exclusively to teach two Internet-related subjects. The students have to create a Web site each and then learn scripting for both client-side and server-side processing, including database connectivity. Each student is given an id and Web space. The server runs Microsoft Internet Information server 5.0 (IIS 5.0) on Windows 2000 advanced server platform. The lab has 20 workstations, each with a unique, hard-wired IP address. These machines primarily run Windows 2000 professional. Students access the server from the special lab, other labs or from outside, using either the university dial-up lines or some ISP. The university routes traffic from ISPs through two proxies. The traffic from on-campus computers and through university dial-up lines is not affected by any proxies. The university semester runs for 16 weeks during which time the students typically complete two assignments, some quizzes and a mini-project each. The lab has been running for almost five years. We chose the latest semester, viz. March-June 2003. Approximately 500 students enrolled in the two subjects.

5.2 Selection of logs

The server logs are created daily, always starting at midnight. We restricted this analysis to a total of 24 days, 10 in April 2003, seven each in May and June. The number of 'hits' recorded by the log ranged from over 4000 to more than 250,000 in a day. The seven days in May and June were deliberately chosen to cover three days before and after the deadline of the assignment that was due on the deadline, when we expected an increasing trend followed by a decline in the number of server hits.

5.3 Data Cleaning

Prior to evaluating the performance of heuristics to identify the users and their activities recorded on the server logs under the current Web log mechanism, data cleaning process should be done to filter out irrelevant log entries. In most cases, only the log entry of the HTML file request is relevant and should be kept for the user sessions. User requests for one URL frequently result in multiple entries in the server logs, independent of one another, representing requests for the hyperlinked elements, such as images, style sheets and so on. Since the main intention of the Web Usage Mining is to get a picture of the user's behaviour, it does not make sense to process such file requests. This also reduces the size of the data to be analysed.

Like most Web log analysis tools, the cleaning process employed in our method performs the following tasks. First, requests for non-HTML URLs are filtered out. Irrelevant items could be identified based on the suffix of the URL name in the log file. For instance, all log entries with filename suffixes such as GIF, JPG, JPEG, gif, jpg, jpeg are ignored. The set of suffixes could be adjusted as needed for particular Web sites by making changes to cleaning criteria. Next, other known useless data is filtered out like entries with particular server response status code such as "401", "403", "404" and "500" which means some error occurred in client or server side. All the entries, which record user id as unknown ("-"), are also removed as wrong ids. Finally, any extra spurious links such as mistyped URLs and spurious agents are removed.

5.4 Terminology

The following terms are used in evaluating the heuristics.

Hits: the number of individual requests to the server. These include, as explained above, requests for not only HTML documents, but also for gif, jpeg etc.

Views: the number of requests to the server after data cleaning is carried out, explained above. This 'cleaned' data is the base to which the heuristic is applied. The total number of views are also analysed using the explicit user ids recorded in the logs.

Sessions: A session is a sequence of page accesses performed by the user to accomplish a task. Sessions are defined on the basis of the amount of time spent on a single page.

Non-HTML hits: As explained above, these arise from a user's request for an HTML page and are superfluous for the present purpose.

Errors: the number of hits that generated error messages from the server.

Spurious hits: the number of hits that did not contain any meaningful URL and/or Agent.

6. Evaluation

6.1 Data Cleaning

Table 1 gives the detail of the effect of data cleaning mentioned before and converts the number of Hits to Views. It covers the logs from 10 days in April. Data cleaning was carried out on the remaining logs as well with a similar pattern of conversion.

Date	Hits	Non-HTML Hits	Error Hits	Spurious agent or Url	Wrong id	Views
1/4	44264	12824	1298	823	325	28994
2/4	97333	24523	3730	1693	557	66830
3/4	31674	8591	1237	426	265	21155
4/4	13384	4470	591	381	138	7804
5/4	5683	1423	177	26	110	3947
6/4	9598	2773	265	61	129	6370
7/4	8884	2147	328	130	264	6015
8/4	13518	3669	845	78	336	8590
9/4	15942	3216	795	114	593	11224
10/4	5448	1012	347	55	211	3823

Table 1 - Data Cleaning: Converting 'Hits' to Views'

It is worth mentioning here that, for the purposes of this paper, we are interested in analysing the number of Views. However, the difference between the Hits and Views is still a demand on the server and could affect the server performance. Further, a greater number of graphic and other images are also indicative of the type of site(s) under scrutiny. It is also a moot point if the number of 'errors' is indicative of the user performance that ought to be taken into account in personalising a Web site.

6.2 User Sessions

Tables 2 to 4 give the details of the performance of the heuristic for the three periods under consideration. The number of Views arrived at after the Data Cleaning step is converted to User Sessions based on the heuristic and also the explicit identification of students in the logs.

The composite heuristic used here works in the following way. First, we use IP/A and Referrer to identify session and then we apply the page timeout of 15 minutes. The number of distinct user sessions is also calculated separately with the help of usr ids. In this case, two separate sessions of the same user are tracked by IP address and Timeout heuristic.

6.3 Analysis

Several points emerge from this analysis. First, the heuristics have over-estimated the number of user sessions on all days. This is to be expected because the numbers of sessions identified through user ids represent near-complete knowledge of the local situation. Heuristics are more generic. The exact number of user ids found in the log of any particular day is the least number of sessions possible. However, an interesting phenomenon came to our notice when we examined the raw (cleaned) data further, viz. that the numbers of user sessions in both cases get inflated for two reasons. The first reason is to do with the use of proxies. The university maintains two proxies to balance the workload of the main servers, which come into operation whenever a student uses an ISP. It can happen that two consecutive accesses from the same student come to the server through different proxies within the time-out threshold of 15 minutes because of load balancing algorithm. The second factor in inflating the numbers is the way the logs are maintained. As mentioned before, each day's log starts at midnight and goes on for the next 24 hours. The logs show that there are a good number of students who work around mid-night. Consequently, when a student starts his/her session before midnight and continues for some time after midnight, that session is split into two logs and is counted twice. To check the extent of such double counting, we merged log files for several days together and ran the analysis again. Tables 5 and 6 illustrate the results. For the 10-day period in April 2003, the number of sessions split across midnights comes to 35. This may not amount to much in the current study but it is indicative of what can happen to the logs, depending upon the policies followed at a given site.

The second point to emerge from this analysis is that the range of differences between the estimates of user sessions by the composite heuristics and those by the user ids is rather large, from just over 8% to more than 100%. This makes it difficult to judge the relative performance at this stage, beyond saying that the composite heuristic 'over-estimate' these numbers. Ideally, one would like to estimate the scale of variation so that the heuristics could be used with more confidence. Further investigation is needed to arrive at more quantifiable understanding..

The third point is about the student behaviour in submitting their assignments, as exhibited by the number of hits and sessions. The submission dates of two assignments were 7th of May and 4th of June and their effect cannot only be anticipated qualitatively but also quantitatively. Tables 3 and 4 reflect these patterns. This has implications for the server and network administrators, the students and the academics in charge. Peak loads can be identified in advance and students can be shown the results of such analyses to help them to submit their assignments relatively trouble-free.

Date	Views	Sessions (heuristics)	Sessions (user ids)	Diff-erence	%Diff
1/4	28994	370	342	28	8.19
2/4	66830	769	654	115	17.58
3/4	21155	316	273	43	15.75
4/4	7804	146	117	29	24.79
5/4	3947	97	74	23	31.08
6/4	6370	117	102	15	14.71
7/4	6015	177	150	27	18.00
8/4	8590	182	167	15	8.98
9/4	11224	274	251	23	9.16
10/4	3823	133	103	30	29.13

Table 2: Performance of Heuristics (April 2003)

Date	Views	Sessions (heuristics)	Sessions (user ids)	Diff-erence	%Diff
4/5	7348	169	133	36	27.07
5/5	9446	353	314	39	12.42
6/5	25896	1370	680	690	101.47
7/5	41663	1399	1012	387	38.24
8/5	9011	297	240	57	23.75
9/5	3315	152	122	30	24.59
10/5	3259	96	77	19	24.68

Table 3: Performance of Heuristics (May 2003)

Date	Views	Sessions (heuristics)	Sessions (user ids)	Difference	%Diff
1/6	53482	857	677	180	26.59
2/6	75853	1514	1228	286	23.29
3/6	207874	2746	2108	638	30.27
4/6	105283	1749	1187	562	47.35
5/6	29786	617	414	203	49.03
6/6	14470	445	289	156	53.98
7/6	7738	255	190	65	34.21

Table 4: Performance of Heuristics (June 2003)

Date (from)	Date (to)	Views	Cummulative sessions (individual run)	Sessions (batch run)	Diff-erence
1/4	2/4	95824	1139	1123	16
1/4	3/4	116979	1455	1429	26
1/4	4/4	124783	1601	1572	29
1/4	5/4	128730	1698	1667	31
1/4	6/4	135100	1815	1784	31
1/4	7/4	141115	1992	1960	32
1/4	8/4	149705	2174	2140	34
1/4	9/4	160929	2448	2410	38
1/4	10/4	164752	2581	2539	42

Table 5: Number of sessions spaned over logs based on the heuristic (April 2003)

Date (from)	Date (to)	Views	Cummulative sessions (individual run)	Sessions (batch run)	Diff-erence
1/4	2/4	95824	996	985	11
1/4	3/4	116979	1269	1250	19
1/4	4/4	124783	1386	1364	22
1/4	5/4	128730	1460	1436	24
1/4	6/4	135100	1562	1538	24
1/4	7/4	141115	1712	1687	25
1/4	8/4	149705	1879	1852	27
1/4	9/4	160929	2130	2099	31
1/4	10/4	164752	2233	2198	35

Table 6: Number of sessions spaned over logs based on user id (April 2003)

7. Conclusions and Recommendation for a Strategy to Evaluate Heuristics

This paper has reported on the investigation into the efficiency of heuristics that may be used to identify user sessions, based on the analysis of Web logs. This was done on the basis of knowing the 'local' circumstances, policies and procedures, which allowed for much better estimates of user sessions. The heuristics, by their very nature, are generic, without this local knowledge. It is still too early to draw definite quantitative conclusions about the efficiency of these heuristics in combination or individually. However, our investigation allows us to recommend a strategy for future work, as outlined below.

1. Start with the log data bearing in mind that it is just the raw data and needs to be 'cleaned up'.
2. Formulate and use a 'Cleaning' procedure. The cleaning procedure will depend upon the local circumstances and policies. Thus, for example, the logs we analysed contained error messages, spurious information, non-HTML requests and wrong (user) ids. It is possible to use more than one log file to log these entries separately. Also, the logs may be hourly, daily, weekly or any time period as determined by the Web administrators. The cleaning procedures must incorporate such knowledge.
3. If any part of the logs contains explicit user ids or session ids, use them to isolate the subsets of data where such local practices will make it easier to understand how much the heuristics vary in their analysis.
4. Analyse the remaining, cleaned data that does not contain any user/session ids, on the basis of the heuristics. Use the understanding from step 3 to refine the analysis, as necessary.

The fourth step in effect use the understanding derived from the third step in an empirical way. That is to say that the user behaviour as understood by carrying out step 3 is expected to remain unchanged.

There is more detailed analysis under way. First of all, we plan to test the statistical significance of each heuristic. Then, various combination of heuristics will be similarly analysed. We have also come across some anomalous results, including the 'outlier' of 101.47% more user sessions on the basis of heuristics (see Table 3). These and more detailed analysis of individual heuristics applied in different situations would be the tasks for the future.

References

1. Pirolli, P., Pitkow, J.E., and Rao, R., *Silk From a Sow's Ear: Extracting Usable Structure from the World Wide Web.* "Conference on Human Factors in Computing Systems (CHI 96)". Vancouver British Columbia, Canada 1996
2. Pitkow, J.E., *InSearch of Reliable Usage Data on the WWW.* "The sixth International World Wide Web Conference". Santa Clara, California 1997
3. Cooley R., Mobasher B., and Srivastava J., *Data preparation for mining world wide web browsing patterns.* "Journal of Knowledge and Information Systems". **1**(1): p. 5-32. Springer-Verlag February, 1999
4. Berendt, B., Mobasher B., Spiliopoulou, M., and Wiltshire, J., *Measuring the accuracy of sessionizers for web usage analysis.* "Proceeding of the Workshop on Web Mining, First SIAM International Conference on Data Mining": p. 7-14. Chicago, IL 2001
5. Berendt, B., Mobasher B., Spiliopoulou, M., and Nakagawa, M., *A framework for the evaluation of session reconstruction heuristics in web usage analysis.* "INFORMS Journal of Computing". **15**(2). 2003
6. *www.w3.org/Daemon/User/Config/Logging.html.* last accessed: 22nd of September, 2003,
7. Greenberg, S. and Cockburn, A., *Getting Back to Back:: Alternate Behaviors for a Web Browser's Back Button.* "Proceeding of the 5th Annual Human Factors and Web Conference". NIST, Gaithersburg, Maryland, USA June 3rd, 1999
8. Tauscher, L. and Greenberg, S., *Revisitation patters in World Wide Web navigation.* "In ACM SIGCHI '97 Proceedings of the Conference on Human Factors in Computing Systems": p. 22-27. ACM Press Atlanta, Georgia, USA March, 1997
9. Shahabi C., Zarkesh A., Adibi J., and V., S., *Knowlege Discovery from Users Web-page Navigation.* "Proceedings of the IEEE RIDE97 Workshop". April, 1997
10. Shahabi C., Banaei-Kashani F., and J., F., *A Reliable, Efficient, and Scalable System for Web Usage Data Acquisition.* "WebKDD'01 Workshop in conjunction with the ACM-SIGKDD". San Francisco, CA August, 2001
11. Elo-Dean, S. and Viveros, M., *Data mining the IBM official 1996 Olympics Web site.* "Technical report". IBM TJ Watson Research Center 1997
12. I-Yuan Lin, X.-M.H., Ming-Syan Chen, *Capturing User Access Patterns in the Web for Data Mining.* "11th IEEE International Conference on Tools with Artificial Intelligence": p. 345. Chicago, Illinois November 08-10, 1999
13. Felciano, R.M. and Altman, R.B., *Lamprey: Tracking Users on the World Wide Web.* "AMIA Annual Fall Symposium". Hanley & Belfus. Washington, D.C. 1996
14. Tak Woon Yan, Matthew Jacobsen, Hector Garcia-Molina, and Dayal, U., *From User Access Pattern to Dynamic Hypertext Linking.* "Fifth International World Wide Web Conference". Paris, France May 6-10, 1996
15. El-Ramly, M. and Strulia, E., *Web-usage Mining and Run-time URL Recommendation for Focused Web Sites: A Case Study.* "Journal of Software Maintenance and Evolution: Research and Practice". **00**: p. 1-7. John Wiley & Son, Ltd 2000
16. Nan Niu, E.S., Mohammad El-Ramly,, *Understanding Web Usage for Effective Dynamic Web-Site Adaptation.* "4th International Workshop on Web Site Evolution". Montréal, Canada October, 02, 2002