A Data Webhouse to monitor the use of a Web Based Higher Education Information System

Carla Teixeira Lopes*, Gabriel David†

* ESTSP/FEUP, Portugal
carla.lopes@fe.up.pt

† INESC-Porto/FEUP, Portugal
gtd@fe.up.pt

Abstract

Monitoring the user behaviour of an information system (IS) is a valuable way to assess its impact and to guide its improvement. In the case of Web Based IS this is usually done analyzing the Web server logs or clickstreams, a too low-level approach with huge amounts of data. Dealing with this calls for data warehouse techniques, which abstract the clickstream to notions like session and navigation path and combine them with information on the structure and meaning of the IS and its users, leading to what is called a Data Webhouse. The paper presents the dimensional modelling of a data webhouse specifically designed to monitor Web-based Higher Education IS and describes some aspects of its ETL process instantiation in SIGARRA, the WBHEIS of the University of Porto.

Keywords: Data Webhousing, Clickstream, Higher Education Information System.

1 Web Based Information Systems Usage Monitoring

The Web is growing in the number of users [10], usage rate [10] and complexity of its sites [4]. The use of this medium as an access interface to organizational information systems (IS) and its applications is also frequent. With the growth of the experience and expectations of the users, the need to know and meet user demands is rapidly increasing. Monitoring how users behave on the system enriches the knowledge about their needs and allows system adaptation based on their previous behaviours [12]. Besides system adaptation, there are other benefits associated with monitoring the use of an IS. In one perspective, it supports the evaluation of the system against its initial specifications and goals. On the other hand, it enables the development of personalization strategies [1, 3, 5], helps increasing the system's performance [5, 6], supports marketing decisions [2], helps on detecting business opportunities that otherwise could remain unnoticed [8] and may contribute to increase the system's security [3, 11].

It is common to see e-commerce websites as an example of websites that may benefit from a data webhouse. The objectiveness and easy measurability of the advantages in this type of sites (usually done in terms of profit) helps to understand it. However, the already mentioned advantages apply to other types of websites, such as Web-based Higher Education Information Systems (WBHEIS).

2 Data Webhousing Systems

One way to monitor the use of Web-based Information Systems (WBIS) is to use data webhousing systems. These systems are fed by a data source that was born with the Web: clickstreams. These Web related streams aggregate information about all the user actions in a website. This is an evolving source that may be enriched with the use of technologies such as XML, which allows a more detailed description of the structure and contents of web pages. Usually, clickstreams are gathered from log files generated by Web servers. The storage of clickstreams and other contextual data in a data warehouse, to understand user behaviour, is what Kimball and Merz [7] called a data webhouse. According to these authors, a data webhouse is crucial to any activity that uses the Web to deliver its services, allowing the improvement of the organization's systems.

Currently, log file analyzers are the most used applications to monitor WBIS usage [9]. Usually, these applications extract data directly from log files and generate several kinds of statistics. However, with this technique it's, if not impossible, hard to obtain the level of analysis possible with data webhouses. Log file analyzers lack the ability to integrate and correlate information from different sources. They can't, for example, correlate the number of accesses from a student to the program he is enrolled to. With a data webhouse this becomes possible, by enriching the information in the database and by using powerful query processors while keeping basic queries simple. Also, log analyzers don't have the ability to find hidden access patterns in log files [6].

3 SIGARRA

SIGARRA is the WBHEIS developed and used by University of Porto (UP). It is installed in 13 of the 14 academic institutions of the University and provides access to pedagogical, scientific, technical and administrative information relevant to each institution.

Although SIGARRA is defined at the institution level and is supported by several database and Web servers, similarities between the WBHEIS`s structure in the several institutions and the nature of a data warehouse suggest the adoption of a centralized architecture at the university level for the data webhouse.

With the exception of the Faculty of Engineering (FEUP), that has its own infrastructures for hosting SIGARRA, all other institutions share the same infrastructures. These are located in IRICUP, an UP central institute. At FEUP, two http servers answer http requests and the database is clustered by the two machines that host the http servers. At IRICUP there are three web servers and the database is on a different machine. At IRICUP there is also a Standby Database. Figure 1 presents the described architecture. In both institutions, all the referred machines are synchronized with a NTP (Network Time Protocol) server.

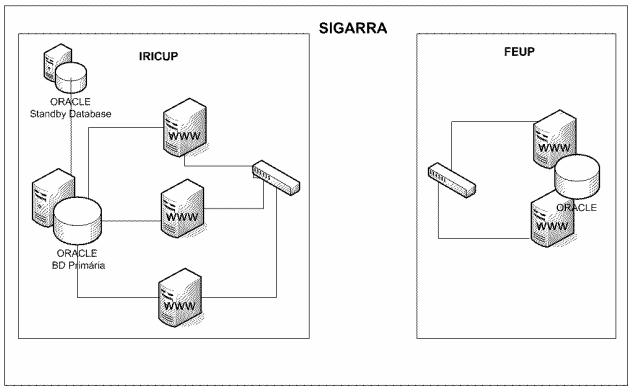

Figure 1: SIGARRA Architecture

4 Dimensional Model

Considering the WBHEIS scenario, a dimensional model has been defined, structuring the webhouse [7] to monitor the usage of this specific type of WBIS. This process has begun with the analysis of the context, followed by the establishment of the granularity, the definition of the relevant dimensions and the identification of the facts.

4.1 The grain of the model

The granularity defines the level of detail that is available in the dimensional model. When choosing the grain, three options have been considered: web hit, web page and web session. The web hit is the most detailed and atomic grain. At this grain each record in the fact table corresponds to an entry in the web server's log file. The second option, the web page, is the aggregation in a single fact of all web hits that are related to the visualization of a Web page (images, css files, etc). The coarser grain is the web session. Adapting the World Wide Web Consortium definition of a server session – a set of clicks, limited in time, of a specific user in a Web server [13] – to the viewpoint of a WBIS, an WBIS session is considered to be a set of clicks, limited in time, from an user in that specific WBIS, disregarding the specific web server that answers the request. According to this definition, in WBIS with multiple Web servers, a WBIS session may be spread by several server sessions.

On one hand, dimensional models should be developed with the most atomic information captured by a business process [14]. The more detailed the facts are, the more information is available and the more powerful is the model. It allows the definition of an increased number of more detailed dimensions and provides a maximum of analytic flexibility because it can be constrained and rolled up in more ways. On the other hand, when the business process is associated with very large quantities of information, it is crucial to choose a granularity that is meaningful to the user and that simultaneously adds value to the organization's knowledge.

In the present warehouse, the main goal is the analysis of user behaviour. So, the web hit granularity is not relevant, as it usually has no meaning in terms of user behaviour, but just in technical level analysis. Accordingly, it has been decided to implement a fact table of Web pages. This grain will answer questions related to user actions inside sessions, which is not possible with just a session fact table. Yet, it also makes sense to have a session fact table, as it allows greater performance on questions related to WBIS sessions. With just a page fact table, these questions would have to be answered indirectly. Therefore, as different grains are not exclusive, it has been opted to include in the model both page and session fact tables.

4.2 Dimensions

After deciding on the grain, the dimensions that enrich the context surrounding facts were identified and characterized. For each dimension, hierarchies, levels and attributes were defined. As seen in Figure 5, the model has 12 dimensions.

Time related dimensions are omnipresent in data warehouses, and this one is no exception. Here, it's crucial to relate time instants and user activities. The **Access Date** dimension is one of the three temporal dimensions in the model. It has only one hierarchy with four levels: Year, Quarter, Month and Day. Each record is a day in the civil calendar. The **Time of Day** is the second temporal dimension. To avoid the size of a dimension that saves the time of day for each day of the civil calendar, it has been decided to split time into a new dimension. This dimension has one hierarchy with three levels: Hour, Minute and Second. Each record is one second of an hour.

The third temporal dimension, specific to the higher education context, is the **Academic Date**. This is an important dimension because it may be used as a conformed dimension - one that may be referred by multiple fact tables, having the same meaning and content in all contexts. An academic calendar is usually associated with different structures that differ on the number of modules. There are structures with two modules (semesters), three (four month periods) and four (trimesters). This dimension has three hierarchies for the semester, four month and trimester

structures and one hierarchy for academic sessions. The latter are specific periods, of variable length, in an academic calendar. For example, the University Day (the UP's anniversary day) is an academic session of one day and the *Queima das Fitas* (a student's celebration week when usually there are no classes) is a session of 7 days. All vacations are academic sessions. The monthly hierarchies have five levels: Year, Module (6, 4 or 3 months), Period (classes, pre-examination or examination period), Week (variable length, defined internally by each faculty) and Day. The academic session hierarchy has three levels: Year, Session and Day. The Session level has information about the start and end date of the session, the session type (with classes, without classes, vacancies) and the number of days in the session.

The **User** is another dimension that keeps information about those who interact with the WBHEIS. This is a crucial dimension to the segmentation of users and to behaviour analysis. Accesses to the WBHEIS can be made by human users (identified or anonymous) or by crawlers. Identified users are students, workers (faculty and staff) and institutions that offer curricular periods of training. Anonymous users are those who access the WBHEIS without signing in.

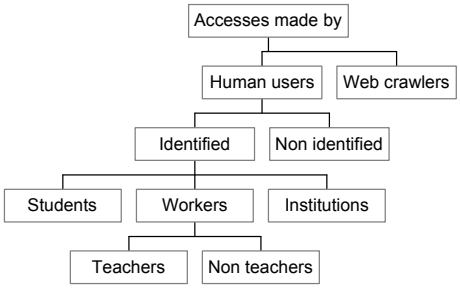

Figure 2: WBHEIS access users

In the Web, obtaining information from users is usually problematic, because Web users often give false information in registration forms having in mind their privacy concerns. Comparatively with other WBIS that gather information from online registration forms, WBHEIS have the advantage of having more trustworthy information about identified users, usually obtained in the student's school registration, in workers act of contract or when institutions offer curricular periods of training. This dimension saves information about the user's academic degree, age group, gender, civil status, activity status, birthplace, role and department/service.

The **User Machine** dimension gathers information about the machine generating the Web request. It has two hierarchies: the physical geography (region, country, domain, user machine) and web geography (top level domain, domain, user machine) of the machine. Besides saving the machine IP address, at the user machine level it is also possible to know if this machine is inside the university, if it is inside the institution and, in this last case, which is the access nature (structured network, wireless network, VPN, etc.).

Another dimension, called **Agent**, keeps information about the agent that has made the request, either a browser used by humans or a crawler. It has two hierarchies: client and platform. The first one has information about the type of agent (text browser, graphic browser, crawler), the agent (for example Microsoft Internet Explorer), the agent's version (6.0) and the log's user agent expression. The other hierarchy has information about the agent's platform (for example Mac, Win, Unix), its specific operating system (for example Windows XP) and the log's user agent expression.

The **Page** dimension is an obvious one when monitoring the use of WBIS. Although this dimension was modelled having SIGARRA in mind, it may be easily adapted to other types of WBHEIS. In SIGARRA all pages are dynamically generated as they are built from database information. Yet, the pages may be classified as static pages (if they only access the database to get the HTML code) or as dynamic pages (if they access tables in the database to get specific data for some areas of a page). In this WBHEIS, there are three concepts associated with a page: application, module and procedure. An application is an autonomous software artefact with one or more modules. These are logical units of the main SIGARRA functionalities and can be seen as a set of related procedures. A procedure generates pages and is the conceptual unit of interaction with the user. The same procedure generates different pages if the arguments it receives are distinct. With this structure in mind, the Page dimension has one hierarchy with four levels: Application, Module, Procedure and Page. At procedure level is saved information about its type (dynamic, static), its access control (controlled access, public access), the languages in which it may generate pages and the id of the help facility. The Page level has the specific URL, the actual language and query.

The **Referrer** dimension describes the page that has preceded the current access. This information is gathered from log files and is related to the domain of the referrer and the referrer itself: port, procedure (if it belongs to the WBHEIS), query (everything that follows the '?' in an URL), the identification and description of the search engine (if this is the case) and the complete URL.

The **HTTP Status Code** dimension has the category of the HTTP Status Code (Informational, Success, Redirection, Client Error, Server Error) and the description of the HTTP status code returned in the HTTP request.

The **Session Type** is a dimension that has not been used from the start but that will be very relevant, as it will allow to aggregate Web sessions into predefined types of sessions. It has only one hierarchy with several levels: session context (for example: the enrolment in a course), the local context (for example: consulting information of a course) and the end state of the session (if its main goal has been achieved).

The **Event Type** dimension has just one hierarchy with one level and it describes what happened in a page at a specific time (for example: open a page, refresh a page, click a hyperlink, enter data in a form).

Information related to the academic institution associated with the Web request is saved in the **Institution** dimension.

4.3 Fact Tables

The page fact table has the finest granularity of the two tables included in the model. This fact table is related to the dimensions represented in Figure 3.

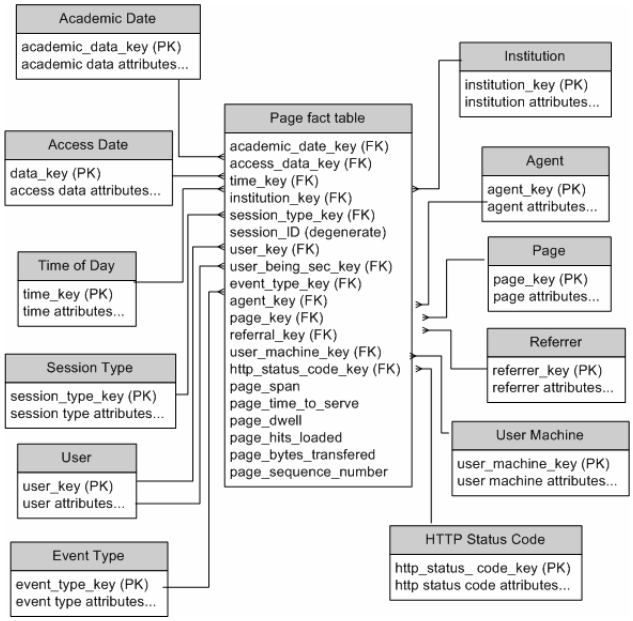

Figure 3: Page fact table

The use of the session_ID as a degenerate dimension, without any relation to a dimension table, allows grouping pages in WBIS sessions. The double connection to the User dimension is due to a SIGARRA's functionality that allows a user to act on behalf of another user (for example, a course grades may be recorded by a secretary instead of the teacher). This fact table has 6 facts that will be described next. The first five are additive ones and the last is a non-additive fact.

Page span is the interval (seconds), between the start of this page request and the next page request. It's t3-t0 in Figure 4.

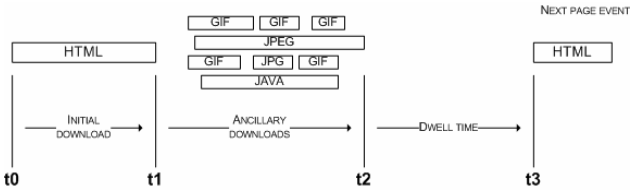

Figure 4: Time intervals in an HTTP Request [7]

Page time to serve is the number of seconds taken by the web server to process all the requests related to this page. It's t2-t0 in Figure 4.

Page dwell is the number of seconds that the complete page is visible in the user's browser. It's t3-t2 in Figure 4. If a page dwell time is too small, it is possible to suspect that there was an error in the loading process or that the page contents were not meaningful to the user. If it's null, the user has stopped page load before it has finished. If it is too large, the user may have been distracted, left the WBHEIS or request pages that were in cache.

Page hits loaded is the number of resources loaded for the presentation of this page.

Page bytes transferred is the sum of the bytes loaded for all the resources related to this page.

Page sequence number is a non-additive fact that represents the sequence number of this page in the overall session.

The session fact table is depicted in Figure 5. The double connection to the Page Dimension allows knowing, more efficiently, which the entry and exit pages are. The temporal dimensions are relative to the session's first request. The referral dimension records the initial referrer of the session.

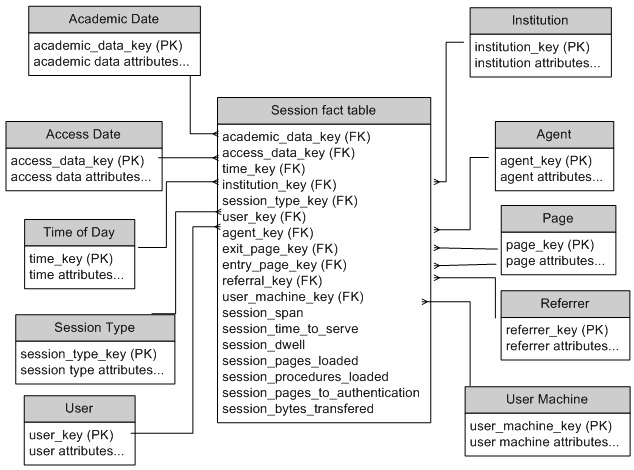

Figure 5: Session fact table

This fact table collects 7 measures. With exception of the *pages to authentication* fact, that is non-additive, all the others are additive.

Session span is the number of seconds between the instant of the first request and the instant of complete load of the last request.

Session time to serve is the number of seconds taken to serve all the requests in the session. It's equal to the sum of the time to serve measure of all the pages in the session.

Session dwell is the total number of seconds of visibility of all the pages in the session. It's the sum of the dwell measure of all the pages in this session.

Session pages loaded is the number of pages in the session.

Session procedures loaded is the number of distinct procedures in the session.

Session pages to authentication is the number of pages until the moment of authentication. If there isn't any authentication, this measure equals session pages loaded.

Session bytes transferred is total number of bytes transferred in this session. It equals the sum of bytes transferred measure of all the pages in the session.

5 Extraction, Transformation and Load Process

The Extraction, Transformation and Load Process (ETL) is a set of processes that feeds the data warehouse. It involves getting the data from where it is created and putting it into the warehouse where it will be used. Although it seems an easy and simple process, this is usually the longest task in any data warehouse project [11].

This section presents some aspects of the ETL process implemented in the SIGARRA context. To apply the methodology in other contexts, the ETL process must be adapted to the new sources in the relevant WBHEIS. However, clickstream data and other data sources, described next, are expected to remain as more or less stable sources.

5.1 Data Sources

As shown in Figure 6, the ETL process has three types of data sources. The **clickstream data** from web servers logs. The data coming from **SIGARRA** which is essential to gather knowledge about the institutions, its internal organization (departments, sections, etc.), academic data (academic calendar, academic events, evaluation periods), the WBHEIS application structure, diverse types of data (countries, councils, parishes, postal codes, etc.) and, most important, knowledge about users which enables clustering and profiling as well as some explanations for detected behaviours. Data from **other data sources** which includes IP ranges of each type of access (wireless, structured network, etc.), data relating IP addresses with geographical areas, domain names, http status codes, and information on search engines, browsers, crawlers, platform and operating system.

5.2 ETL architecture

The general architecture of the ETL process is presented in Figure 6.

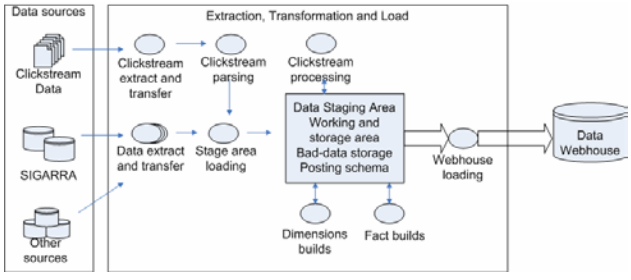

Figure 6: ETL Architecture – based on [11]

As implicit in the name, the extraction phase is the first to occur (**Clickstream extract and transfer** and **Data extract and transfer**). At this phase, all the data sources are moved to the location where they will be processed with a mechanism that may be as simple as a file transfer. Having in attention SIGARRA architecture, in order to minimize data transfers between distinct LANs, the processing area should be at node IRICUP. The clickstream data should be transferred on a daily base, while the other data sources have different transfer frequencies according to their own update frequencies.

After data extraction, the cliscksream data must be parsed (**Clickstream parsing**) and then all data must be loaded into a staging area (**Stage area loading**). The clickstream parsing is done by a PERL program that has a Web log file as input and generates a tab-delimited file with several fields. The steps of this parsing include host IP address resolution, url and referrer parsing, search engine, browser, crawler and operating system identification and cookies parsing.

After all data is loaded into the staging area, clickstream data is processed again (**Clickstream processing**), this time through PL/SQL. Here it is done IP address to country resolution, session, page and user processing, by this order. The session and user processing is done with the help of 3 session cookies (eliminated when the browser is closed), one for non-authenticated sessions, one for authenticated sessions and one for the authenticated user. The first cookie is always present, while the last two are present only when there is authentication. As the first requests in a session are never sent with cookies, this must be analysed through the referrer of these requests in both session and user processing. The session processing must take into account that crawler's accesses are not associated with cookies, what makes impossible session tracking through cookies in these situations. The user processing must also deal with authentications that occur in the middle of a session, where all the requests prior to the authentication must be associated with the user that has signed in.

Although some dimensions (Page, Agent, Referrer and User Machine) can only be built (**Dimensions builds**) after the clickstream processing is done, others can be built at any time (Access Data, Time and HTTP Status Code dimension) and others after the SIGARRA data is loaded into the staging area (Academic Date, Users and Institution). To help build the Page dimension, an auxiliary table is created with all the related information from SIGARRA. The information of this table is then joined with URL clickstream data to generate page instances. The Agent, Referrer and User Machine dimensions are built with information derived from the tab-delimited file generated by the clickstream parsing.

The fact tables are derived (**Fact builds**) after the dimensions due to the dependencies between the two. Hit requests are aggregated into pages and the page fact table measures are calculated. After that, the same is done to the page fact table to obtain the session fact table.

At the end of all the processing, all the records, except those that belong to a not closed session and the last 30 minutes hits are deleted from the staging area. This is due to the fact that a session may not be completed at the time the log file is closed and a new one is started. As a hit that has been served more than 30 minutes later than the previous one isn't considered to be on the same session, only the latest 30 minutes hits and the hits from sessions still not closed may influence the next iteration of the ETL process in the next day.

To make the loading process (**Webhouse loading**) as fast as possible, in the staging area is created a posting schema consisting of tables with the same structure of the warehouse. By the same reason explained above, only the pages from sessions already closed and the sessions already closed are loaded into the posting schema and consequently in the webhouse.

6 Conclusions

In this paper webhouse systems are presented as a solution to monitor the use of Web Based Higher Education Systems (WBHEIS). This solution is presented as the option chosen to monitor the use of SIGARRA, the WBHEIS in use at almost all institutions of Universidade do Porto (UP). The purpose of this paper is to enlarge the application and study area of webhousing systems to the academic context. Despite the similarities between Web based information systems (WBIS), there are important differences between WBHEIS and e-commerce sites that are usually the WBIS used to exemplifications and instantiations of this type of systems.

The paper presents the dimensional model defined to the UP's webhouse, but applicable in other WBHEIS. The Extraction, Transformation and Load (ETL) architecture and its data sources are also described.

Future work involves the data analysis process from which results are expected that may prove the usefulness of these systems, learn more about the user's behaviour and at a later stage help the improvement of the WBHEIS.

References

[1] Jesper Andersen, Anders Giversen, Allan H. Jensen, Rune S. Larsen, Torben Bach Pederson, Janne Skyt. "Analyzing clickstreams using subsessions", *Proceedings of the 3rd ACM international workshop on Data warehousing and OLAP*, pp. 25-32, ACM Press, (2000).

[2] M. S. Chen, J. S. Park, P.S. Yu. "Data mining for path traversal patterns in a web environment", *Proceedings of the 16th International Conference on Distributed Computing Systems (ICDCS 96)*, pp. 385, IEEE Computer Society, (1996).

[3] Robert Cooley. "The use of web structure and content to identify subjectively interesting web usage patterns", *ACM Trans. Inter. Tech.*, 3(2), pp. 93-116, (2003).

[4] Robert Cooley, Bamshad Mobasher, Jaideep Srivastava. "Data Preparation for Mining World Wide Web Browsing Patterns", *Knowledge and Information Systems*, 1(2), (1999).

[5] Magdalini Eirinaki, Michalis Vazirgiannis. "Web mining for web personalization", *ACM Trans. Inter. Tech.*, 3(1), 1-27, (2003).

[6] Karuna P. Joshi, Anupam Joshi, Yelena Yesha, Raghu Krishnapuram. "Warehousing and Mining Web Logs", *Proceedings of the 2nd International Workshop on Web Information and Data Management*, 63-68, ACM Press, (1999).

[7] Ralph Kimball, Richard Merz. "The Data Webhouse Toolkit", *John Wiley & Sons, Inc.*, (2000).

[8] Ron Kohavi. "Mining e-commerce data: the good, the bad, and the ugly", *Proceedings of the 7nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 8-13, ACM Press, (2001).

[9] Richard Li, Jon Salz. "Clickstream Data Warehousing", *ArsDigita Systems Journal*, (2000).

[10] Brij M. Masand, Myra Spiliopoulou, Jaideep Srivastava, Osmar R. Zaiane. "Webkdd 2002: Web mining for usage patterns & profiles", *SIGKDD Explor. Newsl.*, 4(2), 125-127, (2002).

[11] Mark Sweiger, Mark R. Madsen, Jimmy Langston, Howard Lombard. "Clickstream Data Warehousing", *John Wiley & Sons, Inc.*, (2002).

[12] T. W. Yan, M. Jacobsen, H. Garcia-Molina, U. Dayal. "From User access patterns to dynamic hypertext linking", *Computer Networks*, 28, 7-11, (1996).

[13] World Wide Web Consortium. "Web Characterization Terminology & Definitions Sheet", *W3C Working Draft*, (1999).

[14] Ralph Kimball, Margy Ross. "The Data Warehouse Toolkit – 2nd Edition", *John Wiley & Sons, Inc.*, (2002).