RESEARCH PAPERS OF WROCŁAW UNIVERSITY OF ECONOMICS No. 104
Business Informatics 16 2010

Paweł Weichbroth
Technical University of Gdańsk, Gdańsk, Poland
pwi@zie.pg.gda.pl

THE SYSTEM FRAMEWORK FOR PROFILING THE CONTENT OF WEB PORTALS

Abstract: This paper presents a novel approach to profiling the content of Web portals based on discovered association rules from www servers' log files. For this purpose there has been elaborated the agent architecture which is supported by data base and world wide web servers. After a short foreword, in general extent, a description of the system's components is given. Next, the logging process of users' requests to files hosted by the www server is outlined in detail. Then, the functionality of preprocessing, reasoning, dynamic links and the manager agents is characterized. The whole work is finished by conclusions and remarks on future work, in which selected implementation aspects of presented system are pointed out.

Keywords: association rules, Web server, agent framework.

1. Introduction

The dynamic development of the information technology industry caused dissemination of personal computers. Apart from that the buoyant telecommunication industry developed, which resulted in building efficient local, metropolitan and wide computer networks. Today the most famous and used is Internet which roots derive from American Scientific Institutions [Nikodem 2006, p. 161]. The phenomenon of Internet popularity results from adopting developed methods of television and press. Among its services the most successful one is www, meaning public accessibility of the content in the form of Web site.

Web sites can be accessed by means of services defined as portals. The users obtain the anonymous access to those by the Web browser. The www servers are clearly identified by means of the unique name – worldwide domain.

In 2007 the number of the Web sites was estimated for 4 billion connections and every day another million is added [Markov, Larose 2007, p. 4]. The opulence and manifold of the www resources and highly differential level of users' interests encourage for content personalization. Its usage allows for providing better needs fulfillment which results in adaptation of requested information. Described system analyses the usage of the Web sites giving the condition of automatic and intentional modification of information. In the literature such inference is defined as Web usage mining. Commercial usage concern the realm of electronic marketing and trade [Wen 2006].

The paper aims to depict the framework and the functionality of the system developed by the author. The agent framework with the strictly pinpointed division of functionality was adopted for the needs of the system structure. The solutions are based on open source Apache Web server and Postgre SQL implementation.

2. System framework

The presented framework (Figure 1) includes four programme units defined as agents: (1) preprocessing, (2) reasoning, (3) dynamic links, and (4) manager agents, as well as data base server (5) and www server. Those would be discussed in the further part of the paper.

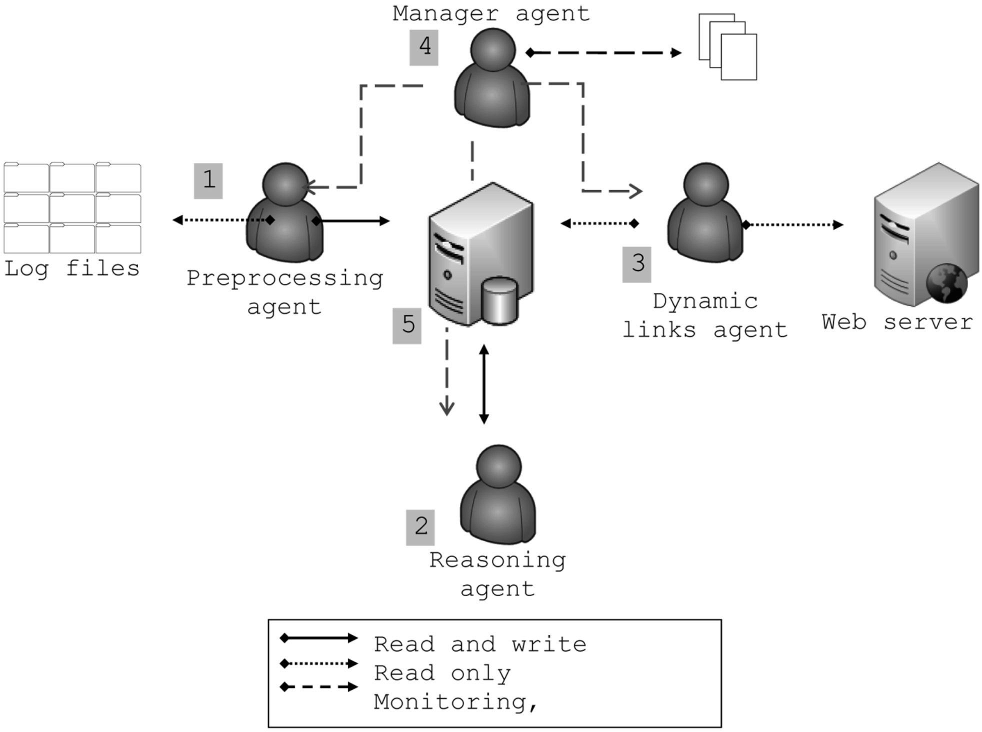


Figure 1. The schema of framework

2.1. WWW server

In the presented framework the www server is the typical one supporting the http protocol [Berners-Lee, Fielding, Frystyk 1995; Fielding et al. 1997, 1999], for example as the one defined by National Center for Supercomputing Applications [NCSA 1996] or popular Apache server [The Apache Software Foundation 2009].

From the users’ perspective, the Web site is unspecified number of html Web pages. The users activity is revealed by opening successive subpages which means clicking on their links. The diversity of the users interests occurs either with the number of opened pages or the time spent on the particular Web page. The schema below presents the session between the user and the server (Figure 2).

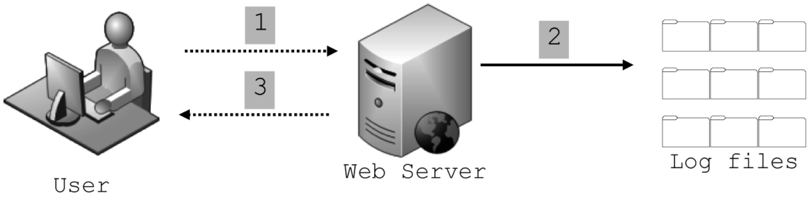


Figure 2. The typical network session between user and www server

Firstly the user, by means of web browser, types the protocol and the name of the server defined as http://nameserver.com (1). Server www saves the user requests in the log files (2) and sends the html document in return (3). The process of logging on the server side will be discussed in details from the perspective of the system framework.

The typical configuration of event logging for the apache server is as follows [The Apache Software Foundation 2009]:

```
LogFormat “%h %l %u %t \”%r\” %>s %b” common
```

Aforementioned configuration registers users requests in the log file known as the Common Log Format (CLF). Such a standard can be used by variety of www servers and read by many applications analyzing the log files. The typical entry in the CLF format [The Apache Software Foundation 2009] is presented below:

```
194.203.88.109 - john [10/Dec/2009:11:35:14 -0700] “GET
/index.html HTTP/1.0” 200 3510
```

where:

194.203.88.109 (%h)	– IP address of the computer client (remote host) which made the request to server,
john (%u)	– identifier of the user who has been authorized and granted file access,
[10/Dec/2009:11:35:14 -0700]	– time in which server accomplished processing the user request, saved in format [day/month/year: hour: minute: second zone],
“GET /index.html HTTP/1.0” (\”%r\”)	– the method used by the client (get or post), the file location meaning the path file and the protocol,
200 (%>s)	– the status code of the file request returned by the server and resend to the client,
3510 (%b)	– size of the file returned to the client request without headers.

Because the connection between the www client and the server in the http protocol is stateless, there is no possibility to keep it after the expiry period. In order to omit the restriction, similarly to [Yan et al. 1996], the session identifier is coded in URL. The new session identifier is generated for the first opened page. This is added to each URL address which represents users requests to the www server files. By this method one id session is kept for many user requests. Additionally, the applied method of session, which expires after specified period of time, ensures that another sessions of the same client are represented by different session id. Among others the Apache server, which is used in system implementation, has such a functionality.

The framework of the given system requires modification of the event logging process running at server side. Default entry format should be supplied by the session id which univocally represents the unique portal user. The modified entry from the server log file altogether with the information including user id ($$45601) is presented below.

```
194.203.88.109 - john [10/Dec/2009:11:35:14 -0700] "GET
/$$45601/index.html HTTP/1.0" 200 3510
```

If the user requested access to *n* pages, the session may be presented as *n*-dimensional vector. For each *i*-th element, we can additionally define weight in the frame of the requested number of particular page, time spent on the page, normalized by the length of the page or the number of the links clicked by the user.

2.2. Preprocessing agent

The main task of the preprocessing agent is to extract selective data from server's log files and to save them in external data base. Information stored in those files, having CLF format, including session id, needs to be cleaned. Those files are read by the agent every specific time period. In the saved data, the unique sessions are searched for and users file requests are attributed to them. Consequently such data are inputted into data base Web Activity Facts (WAF) data base.

2.3. Reasoning agent

This agent, being an intelligent unit of defined tasks, is the core of the system. The tasks are the following:

- frequent itemsets searching,
- knowledge extraction from WAF data base concerning the generation of association rules as well as their saving into Web Activity Knowledge (WAK) data base,
- creating users profiles.

The agent's functioning is mainly based on the implementation of the Apriori algorithm, proposed in [Agrawal, Srikant 1994]. Its precise description can be found in the work [Weichbroth 2009b]. It gives solution to the problem of frequent itemsets searching in large data bases. Originally the research scope consists of five phrases: cleaning, sorting, assigning itemsets, transformation and searching for sequences. In the proposed framework the cleaning phase was intended for the preprocessing agent. In the first place, the agent sorts WAF data base basing on two keys: session identifier and html page path. Next, agent's task is to find all the frequent itemsets, in the respect of the minimum support level. Based on them, agent generates candidate itemsets, joining two one-element into one two-element itemset. The support ratio is counted for each pair – if it is equal or higher than the assumed cut off ratio, such a pair is added to the frequent itemset. In the next step, such a frequent pair will be used to generate three-element candidate itemsets. In sequence, each next step is iterative which means that frequent three-element itemsets will be used to create four-element candidate itemsets, frequent four-element itemsets to create five-elements candidate itemsets and so on [Weichbroth 2009a].

In the context of conducted research [Mikulski, Weichbroth 2009; Weichbroth 2009a], precisely defined path of html file is the one which represents the item. The analysis of log files discovered interesting Web user profile which can be represented by a vector [index.html; biznes.html; info.html] [Mikulski, Weichbroth 2009]. The results of Web usage mining of different Web sites can also be found in [Hatonen et al. 2003; Ivancsy, Vajk 2006; Kosala, Blockel 2000].

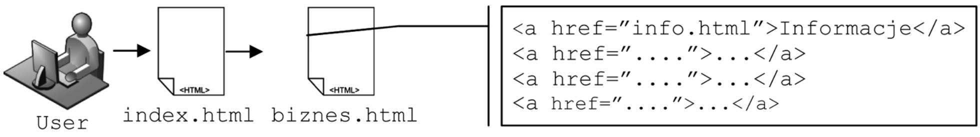
2.4. Dynamic link agent

As mentioned above, discovered users profiles are stored in the WAK data base. The agent classifies user which means assigning him to the profile basing on the files requests.

Active user's vector represents only partial entry of the current session provided that another Web pages will be opened. In order to classify the user, the distance between the profile median and the partial vector is not a good matching measure [Yan et al. 1996]. From this point of view, it is obvious that such a vector has got more zero elements in comparison to the median vector. The solution of this problem is to define arbitrarily the activity threshold, which means the number of accessed pages by the user. If such defined threshold is reached (for example two pages), an attempt is made to match partial vector of current user session to one or more profiles.

After assigning the current user session to one or more profiles, the agent checks those pages whose access was refused. Links to these pages will be placed at the top of the page (Figure 3).

Because the user can have different information needs each time they visit the Web site, the content is each time generated basing on the unique session identifier. To illustrate the idea of the agent, the aforementioned example of the profile will be recollected. On the biznes.html web page the link to the info.html Web page and others related links will be displayed firstly.



```
<a href=”info.html”>Informacje</a>
<a href=”....”>...</a>
<a href=”....”>...</a>
<a href=”....”>...</a>
```

Figure 3. The effect of affecting the agent to assigned users profile

Information about users’ requests to html pages are stored in the log files. Consistently the agent matches sessions to the discovered users’ profiles. In case of discrepancy between current session with any known profile, the agent suspends the matching process and goes to the idle state. The agent is resumed when the counter will be equal or exceeds the activity threshold. This way the agent, for the partial session vector, attempts again to match it to the discovered profiles.

2.5. Manager agent

The reason of the presented framework is indentified, multidimensional functionality which the system requires. The division into agents is motivated by distinguished functionality, assigned to every each of them, independently of one another.

The manager agent’s functions to the resisting agent in the system are listed below:
- receiving messages about jobs executing,
- receiving messages about the status of pending jobs,
- receiving messages about the status of finished jobs,
- monitoring activity state,
- generating reports.

3. Conclusions and future work

This work was aimed to present the architecture and functionality of the system designed to the automatic profiling of the Web server resources users. Tasks division into agents had several objectives. The Main cause concerns the separate and independent functionality of each agent. On the other hand, it provides higher system reliability. Each agent is represented by an isolated process in the operating system. This involves the assignment of a unique range of memory addresses and thread allocation of CPU time. Choosing and implementing a Web server and database management system from the open source solution is obvious. Firstly, access to source code is possible, and secondly, it can be developed with the necessary functionality, regardless of the software vendor. At the current level of proposed solution, the implementation of reasoning agent was made in object-oriented language Java [Mikulski, Weichbroth 2009]. The studies [Mikulski, Weichbroth 2009; Weichbroth 2009a] showed the effectiveness of the algorithm and indicated the direction for further research.

References

Agrawal R., Srikant R. (1994), Fast algorithms for mining association rules, [in:] *Proceedings of the Twentieth International Conference on Very Large Data Bases*, Morgan Kaufmann, San Francisco, pp. 487-499.

Berners-Lee T., Fielding R., Frystyk H. (1995), *Hypertext Transfer Protocol – HTTP/1.0. Internet Draft*, http://www.w3.org/Protocols/HTTP/1.0/draft-ietf-http-spec.html.

Fielding R., Gettys J., Mogul J., Frystyk H., Berners-Lee T. (1997), *Hypertext Transfer Protocol – HTTP/1.1*. Internet Official Protocol Standards (RFC 2068), http://tools.ietf.org/html/rfc2068.

Fielding R., Gettys J., Mogul J., Frystyk H., Masinter L., Leach P., Berners-Lee T. (1999), *Hypertext Transfer Protocol – HTTP/1.1*. Internet Official Protocol Standards (RFC 2616). http://www.w3.org/Protocols/rfc2616/rfc2616.html.

Hatonen K., Boulicaut J.F., Klemettinen M., Miettinen M., Mason C. (2003), Comprehensive Log Compression with frequent patterns, DaWaK 2003, *Lecture Notes in Computer Science* 2737, Springer-Verlag, Berlin, pp. 360-370.

Ivancsy R., Vajk I. (2006), Frequent pattern mining in web log data, *Acta Polytechnica Hungarica*, Vol. 3, No. 1, Budapest, pp. 77-90.

Kosala R., Blockel H. (2000), Web mining research: A survey, [in:] *Newsletter of the Special Interest Group (SIG) on Knowledge Discovery and Data Mininig SIGKDD*, GKDD Explorations 1, Washington.

Markov Z., Larose D.T. (2007), *Data Mining the Web. Uncovering Patterns in Web Content*, *Structure and Usage*, John Wiley & Sons, New York.

Mikulski Ł., Weichbroth P. (2009), Discovering patterns of visits on the Internet web sites in the perspective of associative models, *Polish Journal of Environmental Studies*, Vol. 18, No. 3B, Olsztyn, pp. 267-271.

NCSA HTTPd Development Team (1996), *NCSA HTTPd*, http://hoohoo.ncsa.illinois.edu/.

Nikodem R. (2006), Technologie sieciowe i komunikacyjne, [in:] *Informatyka ekonomiczna. Część I. Propedeutyka informatyki. Technologie informacyjne*, Ed. J. Korczak, Wydawnictwo Akademii Ekonomicznej we Wrocławiu, Wrocław, pp. 159-190.

The Apache Software Foundation (2009), *Apache HTTP Server Version 2.2 Documentation*, http://httpd.apache.org/docs/.

Weichbroth P. (2009a), Analiza zachowań użytkowników portalu onet.pl w ujęciu reguł asocjacyjnych, [in:] *Inżynieria wiedzy i systemy ekspertowe*, Eds. A. Grzech, K. Juszczyszyn, H. Kwaśnicka, N.T. Nguyen, Akademicka Oficyna Wydawnicza Exit, Warszawa, pp. 81-88.

Weichbroth P., *Odkrywanie reguł asocjacyjnych z transakcyjnych baz danych*, [in:] *Informatyka ekonomiczna* 14, Ed. A. Nowicki, Prace Naukowe Uniwersytetu Ekonomicznego we Wrocławiu nr 82, Wrocław 2009b [forthcoming].

Wen J.R. (2006), Enhancing Web search through query log mining, [in:] *Encyclopedia of Data Warehousing and Mining*, Ed. J. Wang, Idea Group Reference, Hershey, pp. 438-442.

Yan T.W., Jacobsen M., Garcia-Molina H., Dayal U. (1996), From user access patterns to dynamic hypertext linking, *Computer Networks and ISDN Systems*, Vol. 28, No. 7-11, Amsterdam, pp. 1007--1014.

ARCHITEKTURA SYSTEMU PROFILOWANIA TREŚCI WITRYN INTERNETOWYCH

Streszczenie: w artykule zaprezentowano nowe podejście do profilowania treści witryn internetowych na podstawie odkrytych reguł asocjacyjnych z plików loga serwera www. Do tego celu opracowano architekturę agentową, która współpracuje z serwerami baz danych oraz www. Po krótkim wstępie, w ogólnych ramach, zostały opisane komponenty systemu. Następnie szczegółowo nakreślono proces logowania żądań użytkowników do plików, udostępnianych przez serwer www. W dalszej części pracy scharakteryzowano funkcjonalność agentów przetwarzania wstępnego, wnioskowania, dynamicznych linków oraz menedżera. Całą pracę zamyka zakończenie, w którym wskazano wybrane aspekty implementacji przedstawionego systemu.