Building and Evaluating Non-Obvious User Profiles for Visitors of Web Sites

Naveed Mushtaq
Karsten Tolle

Peter Werner
Roberto Zicari

Databases and Information Systems (DBIS)
Johann Wolfgang Goethe-University of Frankfurt
Robert-Mayer-Strasse 11-15, D-60325 Frankfurt,
Phone: +49-69-798-28823
Germany
{mushtaq, tolle, werner, zicari}@dbis.informatik.uni-frankfurt.de

Abstract

Building profiles of registered users of a web site, as in case of a portal, is of importance if it goes beyond collecting the obvious information the user is willing to give at the time of registration. The starting point of this work is the assumption that a community of users is registered on a web site and that for each user a profile is built. Such a profile contains two parts of data: one obvious, given directly by the user and one less obvious, inferred by the user's behavior during his visits on the site.

This paper presents ideas on how to build a user profile based on non-obvious information and takes into account various factors influencing in its development. Special emphases are paid on feedback mechanism and several methods of measuring its results.

Keywords: *Non-obvious user profile, Web page ranking, Feedback mechanism*

1. Introduction

The acceptance of a web site, by registered users, for instance a portal, is dependent on how it is valued by them individually.

This value can be raised by, building detailed user profiles, which go beyond collecting the obvious information; the user is willing to give when registering at the web site.

For our work, we assume that a community of users is registered at a web site and that for each of them a detailed profile is built. A profile contains two parts:

- one obvious, given directly by the user (e.g. name, address, e-mail, etc.)
- one less obvious, inferred by the user's behavior in one or more sessions on the web site

Our objective is to develop a realistic user profile based mainly on *non-obvious information*.

This paper presents ideas on how to build a user profile based on non-obvious information and takes into account various factors influencing in its development. Special emphases are paid on feedback mechanism and methods of measuring its results.

The paper is structured in four parts. Section 1, is an introduction to the concept, section 2, concentrates on methods of computing *non-obvious user profile (NOP)*, including usage of algorithm. Section 3 elaborates on Feedback Mechanism and describes its role in building *non-obvious user profiles*. Section 4 relates to planned future work, in this area.

2. Building non-obvious user profiles

A web site (or a domain) is a collection of web pages that are linked together within the site. Often these pages are also linked to external pages outside the domain). Each page has a specific content (in our model this relates to a specific topic). This link structure can be represented by a directed graph where the nodes are the web pages and the vertices are the links.

Until now, we assume that the web site is static that means that there are no changes in it, neither in the structure of the graph or in the content of the individual pages. Dynamic changes will be discussed in a separate paper, like mentioned in section 4.3. The main difference between the pages inside the web site (domain) and the external pages is that the internal ones are in control of the site's web server and the content they represent is clear to the owner; where as the external pages are not.

2.1. The algorithm for building non-obvious user profiles

In this section we present a basic algorithm to build *non-obvious user profiles,* in the rest of this paper denoted as *NOP*s.

The relevant data to be used in the algorithm is extracted from various log files generated by the web server.
The basic algorithm we are experimenting for building *non-obvious user profiles* works as follows:

1. The owner of the web site defines statically a list of key topics $Tp_1, Tp_2, Tp_3, \ldots, Tp_n$. Each of which corresponds to a certain area of interest.
2. The owner of the web site statically creates a correspondence between the list of topics with the *TOPIC_ID*s $Tp_1, \ldots, Tp_n$ and the current set of pages with the *PAGE_IDs* $P_1, \ldots, P_m$ the web site is composed of. It is possible, that a page P_j is connected to several topics. This is realized by introducing a *WEIGHT* $v_j(Tp_i)$ ($0 \leq v_j(Tp_i) \leq 1$) that represents the strength of the relation between the page P_j and topic Tp_i.
   Each page of the web site has therefore an association to a set of topics, as a proper subset of the total list of topics.
3. The *non-obvious user profile* (*NOP*) of each registered user contains the actual number of his sessions *SCOUNT* and a list of weights $w_1, \ldots, w_n$ representing his interest in the key topics $Tp_1, \ldots, Tp_n$. These are normalized to a value between *0* (no interest) and *1* (very strong interest) or could contain a *NULL*-value, if no information is available.
4. After each session of a certain registered user, his interests in the topics $Tp_1, \ldots, Tp_n$ during the actual session are calculated and stored in the variables $x_1, \ldots, x_n$. This is realized by the following rule that includes the total duration of the actual session in the nominator and the time the user spent on the different web pages related to the topics $Tp_1, \ldots, Tp_n$.

$$x_i = \frac{\sum_{j=1}^{m} duration(P_j) \cdot v_j(Tp_i)}{\sum_{k=1}^{m} duration(P_k)}$$

   Where duration(P_j) indicates the time spent visiting page P_j.
   We can see that x_i is normalized by the total duration of the actual session to a value between *0* and *1*.
5. At the end of a session the calculated values $x_1, \ldots, x_n$ associated to the various topics will be stored in the user profile. It is important to include the old, stored value w_i, but also it is obvious that the current session is more interesting than the older ones. This is realized by introducing a factor *f*. The value for this factor depends on the application and the circumstances.
   The new values for the user profile are calculated by the following rule:

$$w_i = \frac{scount \cdot w_i + f \cdot x_i}{scount + f}$$

In this way, the *non-obvious user profile* (*NOP*) of each registered user is kept current based on his behavior on the web site. This means the profile is only valid for a certain point in time, we therefore keep the timestamp of the last time the profile was updated within it.

2.2. Introductory example

To explain the algorithm we present a simple example. We assume that a web site contains three pages that are specific to four topics. This is represented in figure 2.1.

Page 1 is comprised of all topics with the same strength, page 2 only contains information about topic 1 and page 3 relates to the topics 1, 3 and 4 and has a special interest in topic 4 since its *weight* of *0.60* is much higher than for the topics 1 and 3 having a *weight* of *0.20*.

PAGE_ID	URL
1	http://www.Otland.de/index.html
2	http://www.Otland.de/xml.html
3	http://www.Otland.de/metadata.html

TOPIC_ID	LABEL	DES
1	XML	EXtensilble Markup Language
2	Profiles	Profiles for Users visiting a Web site
3	RDF	Resource Description Framework
4	SOAP	Simple Object Access Protocol

PAGE_ID	TOPIC_ID	WEIGHT
1	1	0.25
1	2	0.25
1	3	0.25
1	4	0.25
2	1	1.00
3	1	0.20
3	4	0.60
3	3	0.20

Figure 2.1: Example – Pages of a website with their topics and weights

The *non-obvious user profiles* are represented in their table, containing the number of sessions and the degree of interest a user has in different topics like mentioned in figure 2.2.

USER_ID	SCOUNT	TIMESTAMP	Tp_1	Tp_2	Tp_3	Tp_4
1	1	05.09.2003 16:15:56	0.3	0.3	0.0	0.6
2	2	12.09.2003 17:12:42	0.4	0.2	0.9	0.7
3	1	07.09.2003 13:19:58	0.1	0.3	0.5	0.2
4	3	06.09.2003 12:55:01	1.0	0.0	0.0	0.0

Figure 2.2: Example – *Non-obvious user profiles* of four different users with their

As can be seen in the above table, the users 1 and 3 had only one session, user 2 had two sessions and an interest in different topics and user 4 had three session and indicated interest only in topic 1.

User's activity during a session on the site is extracted from the web log file, utilizing an existing log analyzer tool or manually. Typically, information in log files is extensive, but for our purpose, only the relevant data is retrieved and stored in a table of a database (figure 2.3).

TIMESTAMP	PAGE_ID	DURATION	SESSION_ID	USER_ID
05.09.2003 12:09:23	1	10	1	1
05.09.2003 12:09:33	2	10	1	1
05.09.2003 12:09:43	3	20	1	1
05.09.2003 12:10:03	2	10	1	1
05.09.2003 16:15:46	1	10	2	1
05.09.2003 16:15:56	3	10	2	1

Figure 2.3: Example – Database table with

The stored data on this table reflects a user's activity on the web site, as can be seen in figure 2.3. The user having the USER_ID 1 visited in his first session a few pages, represented in the click stream *(1 2 3 2)* and in his second session he left the click stream *(1 3)*. Several sessions are filtered out and are then used to calculate the user's interest during these sessions as mention earlier in the algorithm stated above. Later the result is included in the user profile and the table in figure 2.2 is altered.

3. Measuring the results

The algorithm we presented in the last section allows us to include the user's behavior on the web site into his *non-obvious user profile (NOP)*. The *NOP* is improved gradually with each visit on the site (session).

3.1. Including a feedback mechanism

An important question at this point is to understand if "patterns of interests" change or remain constant for each users visiting the web site. This information can be very valuable for the owner of the web site. It would be desirable to have a notion of "accuracy" for a *NOP*, or at least to be able to observe and measure changes in time of the *NOP* and how the calculated *NOP* relates to the explicit declaration of interests the user might be willing to give.

To measure the accuracy of a *NOP* we use a feedback mechanism, where the user is asked directly to enter his preferences, e.g., by presenting the values of the *NOP* and asking the user to verify or correct them. From the feedback given by the user we build up another profile called the *feedback profile (FP)*.

For an optimal comparison of the *NOP* and the *FP*, the *NOP* is updated with the latest session information as described in the previous section together with the start of the feedback mechanism. We can therefore say that both profiles (*NOP* and *FP*) are valid at this point in time t_i. The profiles are then called $NOP(t_i)$ and $FP(t_i)$ for a certain user.

In figure 3.1 a *FP* for the user with USER_ID 2 is shown that corresponds in time to the *NOP* shown in figure 2.2 for this user. We assume to have at least access to the *NOP* and *FP* generated during the previous time the feedback mechanism was started called time t_{i-1}.

USER_ID	TIME_STAMP	Tp_1	Tp_2	Tp_3	Tp_4
2	12.09.2003 17:12:42	0.5	0.3	0.0	0.9

Figure 3.1: Example – Feedback profile *FP* of user with the *USER_ID* 2 at time t_0

To prevent from being a source of annoyance to the user, the feedback mechanism is not activated every session. There are various ways in which the feedback mechanism can be started, for example it can be triggered on certain events or can be activated after a change is detected.

We use these four profiles $NOP(t_{i-1})$, $NOP(t_i)$, $FP(t_{i-1})$ and $FP(t_i)$ for a basic measurement, resulting in three variables (*D*, *FD* and *ND*), defined later below, that can be observed over time to find out how close the generated *NOP* is to the real interests of the user as entered in *FP* and also taking into account, that the pattern of interests for a user will change over time. The idea of calculating these variables is visualized in figure 3.2.

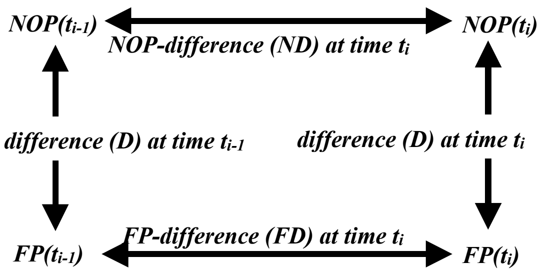

Figure 3.2: Visualization of the variables *D*, *FD* and *ND*

These three values are calculated for each observed topic separately and defined as follows:

- ***Difference***, denoted as ***D(Tp$_j$)*** – where we compare for each topic Tp_j the current *NOP* with a current given *FP* at a certain point of time.

  So this value represents the difference between the computed *NOP* and the entered *FP* of a certain user. For a point in time t_i it is calculated by:

  $$D(Tp_j,\, t_i) := FP(Tp_j,\, t_i) - NOP(Tp_j,\, t_i)$$

- ***FP-difference***, denoted as ***FD(Tp$_j$)*** – where we compare for each topic Tp_j the current *FP* with the last value of *FP*, stored in the database.

  So this value represents the change of interest of a user in a certain topic, based on its entered feedback. For a point in time t_i it is calculated by:

  $$FD(Tp_j,\, t_i) := FP(Tp_j,\, t_i) - FP(Tp_j,\, t_{i-1})$$

- ***NOP-difference***, denoted as ***ND(Tp$_j$)*** – where we compare for each topic Tp_j the current *NOP* with the last value of *NOP*, stored in the database.

  So this value represents the change of behavior of a user, based on its click-streams. For a point in time t_i it is calculated by:

  $$ND(Tp_j,\, t_i) := NOP(Tp_j,\, t_i) - NOP(Tp_j,\, t_{i-1})$$

Since each single value of the profiles *NOP* and *FP* are between *0* and *1*, the values for the three variables *D*, *FD* and *ND* above will be between *-1* and *+1*, which means they also reflect the *direction* of the change. For example based on the values represented in figure 3.3, the value $FD(Tp_2,t_1)=+0.2$ would mean a rising interest of the user for topic Tp_2 at the time t_1 compared with the time t_0 and the value $FD(Tp_2,t_2)=-0.4$ describes a decreasing interest from the time t_2 to the time t_1.

FP	Tp1	Tp2	Tp3	Tp4
t_0	0.5	0.3	0.0	0.9
t_1	0.5	0.5	0.0	0.1
t_2	0.5	0.2	0.0	0.1
t_3	0.5	0.4	0.0	0.1

NOP	Tp1	Tp2	Tp3	Tp4
t_0	0.4	0.2	0.9	0.7
t_1	0.45	0.3	0.7	0.7
t_2	0.6	0.2	0.8	0.3
t_3	0.5	0.3	0.7	0.2

Figure 3.3: Example – Changes of the *NOP* and *FP* of the user 2 from time t_0 to t_3

The values given in figure 3.3 will result in the graphs for the three variables, represented in figure 3.4.

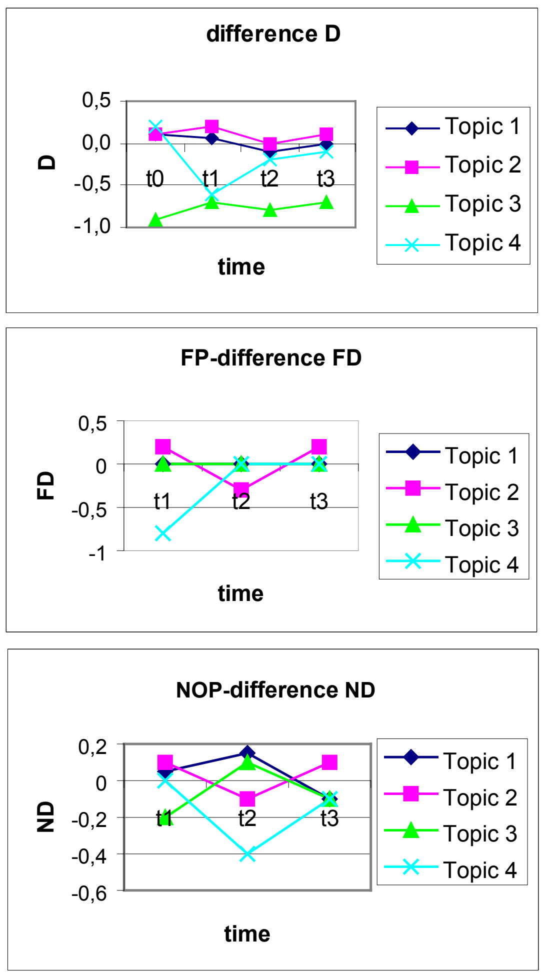

Figure 3.4: Example – Diagrams for the changes of *D*, *FD* and *ND*

3.2. Using the feedback information

The natural question at this point is what to do with all these measured values. Our ultimate goal is to provide the owner of the web site with enough information to make decisions. In this respect, measuring the various *D*, *FD* and *ND* over time is, in analogy, like a airplane pilot who reads from his cockpit all kinds of measured variables (e.g. current altitude, wind, etc.) in order to make a decision, for example manual landing, or instead insert the automatic pilot.

Having the additional entered information from the feedback given by the user in *FP*, this information can additionally be used to improve the future *NOP*. This can be achieved in various ways having different advantages or drawbacks.

We would like to stress here, that we are not trying to "interpret" or question the feedback given by the user, but simply to offer an additional information to the owner of a web site so that he can make a qualified decision, as illustrated inn the various alternatives below.

Here we present three possible alternatives:

<u>Alternative 1:</u> When a feedback is given, the *FP* could be used as the new base for generating future *NOP*s. This method has the advantage that changes of interests will have a very strong impact on future *NOP*s and therefore the *NOP* will be closer to the current interest of the user. On the other hand the user might give wrong answers, either on purpose or by having a different understanding of the topic name. This would result in a wrong starting point for next *NOP*s.

<u>Alternative 2:</u> The *FP* has no influence at all on the creation of the *NOP,* which means we use the *NOP* without change after a feedback is given for the creation of future *NOPs*. We then would avoid wrong answers or misunderstandings, but changes of interest would be realized much later in future *NOP*s and we don't benefit from the user's feedback.

<u>Alternative 3:</u> We could generate a *derived profile (DP)* as the new starting point for future *NOP* generation. The *DP* could be calculated as defined by a set of rules defined for example by the owner of the web site. In this way, the owner can choose which decision to take based upon the values of the measured variables.

Here we show an example of this alternative. Let us assume that the owner of the web site has defined the following rules:

(Rule 1) $if\ |FD(Tp_j, t_i)| > y\ AND\ \ 0<y<1 \Rightarrow$
$DP(Tp_j, t_i) := FP(Tp_j, t_i)$

If the feedback profile *FP* changes more than a given value *y*, this indicates a fundamental change of interest of the user, and this will be 100% included into the derived profile *DP* by just coping the value. The threshold value *y* is given by the owner of the web site.

(Rule 2) $if\ |D(Tp_j, t_i)| < x\ \ AND\ 0<x<1 \Rightarrow$
$DP(Tp_j, t_i) := avg(FP(Tp_j, t_i), NOP(Tp_j, t_i))$

If the difference between the calculated *NOP* and the entered *FP* is smaller than a given value *x*, the derived profile *DP* is the average of both values. This offers a balance between a slight change of interest and an unequal understanding for the interpretation of the given values. The threshold value *x* is given by the owner of the web site.

(Rule 3) $else \Rightarrow$
$DP(Tp_j, t_i) := NOP(Tp_j, t_i)$

If neither Rule 1 nor Rule 2 applies, we assume a stable pattern of interest (since $FD < y$) but also a significant difference between the given feedback *FD* and the user's behavior, represented by his *NOP* (since $D > x$). There can be different reasons for this situation, a wrong feedback given by the user, a complete different semantic understanding of the topic, a wrong topic-page mapping etc. In this case the last given behavior stored in the *NOP* is used for the new *DP*. But this should be done with care. A system implementing it should try to use additional information to find out the reason for this situation. E.g. by recognizing that most of the users are ending up in the third rule, this could indicate, that there is a wrong topic-page mapping or a fundamental semantic mismatch.

This simple example shows that by using the measured variables and offering the possibility to program your own rules, it is possible for the owner of the web site to build up a set of rules that govern the construction of *non-obvious user profiles*, both taking into account the user feedback, the user behavior and the owner's (changing) goals.

In general, depending on the alternative above chosen, the *NOP* will change accordingly. Therefore it also affects the current values for *D* and *ND*, while the value for *FD* stays the same.

There are further ways to handle the feedback given by the user, especially if additional information can be included. Which way to choose might also depend on the specific domain where the system is used and what the user wants to achieve from it.

4. Future work

There are a lot of additional ideas and possibilities to continue the work on building *non-obvious user profiles*. Some of them are mentioned in this section.

4.1. Cluster the non-obvious user profiles

The profiles that are built could be used to group similar users in a cluster. This would give us new possibilities in improving the service for the users. Standard data mining tools would help to cluster the profiles. This allows us to recognize additional information like cross-selling etc. The collected *non-obvious user profiles* can be used to support classical personalization strategies.

These clusters can help to find pattern and therefore to understand better the feedback given by the user described in section 3.1. This additional knowledge could be used to run a more advanced rule system as that we presented as Alternative 3 in section 3.2.

4.2. Improving the algorithm

In some scenarios, it makes sense to introduce a ranking of (internal) pages for the web site. That means, that each page P_j is weight by the owner (with a value o_j between 0 and 1) corresponding to its importance. For example in a web site of an online store the page with a buying functionality would be more important than a page with some content about a product information. This value has to be entered by the owner of the site or has to be calculated by a special algorithm like *PageRank* introduced by Brin, Page et.al. [3][4]. At the moment, we assume that the owner of the site enters the weights manually.
In future approaches, we want to use this general page ranking to improve the quality of the *non-obvious user profiles* to include the site owner's opinion about the relevance of the pages.

4.3. Static versus dynamic profile building

The paper assumed that the web site, both in content and structure is static. Of course, this is a major limitation. We will look in a separate paper at the issue of dynamic changes of a web site, both content and structure and the issue of building user profiles in this environment. In a separate paper we will analyze changes in the list of topics associated to the web site, the page content of the web site, the structure of the web site and the structure of the external environment to the web site.
In particular, the algorithm to calculate *NOP*s will have to be changed accordingly. Dynamic changes also have an impact on the way measurements are done.

4.4. Matching the topics

To bring together the pages of a web site and a list of corresponding topics is a task that is currently done manually in our prototype. For each page a list of corresponding topics and their weights are entered in a database table (see figure 2.1).
But there are methods by which the content of a page is identified and it's semantic extracted automatically. We could for example assume that the pages themselves contain annotated metadata describing the content in order to extract automatically the topics from the web pages This is subject of a future work, where we will look at the issue of using RDF-like notation to define meta-data [15].

5. Related work

The importance of building user profiles is obvious and there is a lot of related work in this area.
Yan, Jacobsen, Garcia-Molina and Dayal [16] describe an approach for automatically classifying visitors of a web site according to their access pattern. User access logs are examined to discover clusters of users that exhibit similar information needs, and based on categories of users with similar interests they dynamically suggest new links. This leads to a better organization of the hypertext documents for navigational convenience.
Similarly a system like Letizia [11], records what interests a user has shown and then suggests new pages that could be relevant for a class of users with similar interest.
WebWatcher [10] proposes a learning approach to provide navigation hints. User feedback is used to improve the quality of the hints.
Our approach is different. We analyze the user's behavior using a page ranking algorithm, and we concentrate more on defining a user feedback and a notion of relevance of the calculated results, in order to measure "change of interests" patterns.

Acharyya and Ghosh [1] propose a framework for modeling users whose surfing behavior is dynamically governed by their current topic of interest. D'Ambrosio, Altendorf and Jorgensen [2] suggest probabilistic relational models of online user behavior. Both approaches use advanced algorithms for calculating the user's behavior, but they do not include a feedback mechanism and do not define a notion of relevance of results.

The first and most often used algorithm for page ranking was developed by Brin and Page [3] [4]. This algorithm could be adapted and used to improve the quality of our *non-obvious user profiles* as mentioned in section 4.2.
Haveliwala [7] extends the page ranking algorithm of Brin and Page, by including search queries. This results in a context-sensitive ranking corresponding to the user's request.

Casteleyn, De Troyer and Brockmans [6] argue, that the change of web sites, they call it “adaptive web sites”, must be controlled to keep the overview. They designed an adaptation specification language to configure runtime restructuring of web pages. Their ideas could be interesting for further implementations of our system.

Hay, Wets and Vanhoof [8] introduced an algorithm called Sequence Alignment Method for clustering navigation patterns on a web site, using a so called Interestingness Measurement [9] including structure and usage data. Their ideas could be helpful for further applications based on our *non-obvious user profiles*.

Acknowledgements: The results of this work is part of the SIMAT project [5], supported by the European Commission, as a trial project, part of the EUTIST cluster on Agent Middleware Infrastructure Technologies under grant number IST-2001-32432. We would like to thank you the following colleagues for their constructive feedback on earlier versions of this paper: Ciarán Bryce for reviewing the algorithm and giving us useful feedback, and Georg Gottlob who encouraged us to work on the user feedback problem. We would also thank you the reviewers of this paper for their useful comments.

6. References

[1] S. Acharyya, J. Ghosh : Context-Sensitive Modeling of Web-Surfing Behaviour Using Concept Trees, WEBKDD 2003, 2003

[2] B. D’Ambrosio, E. Altendorf, J. Jorgensen: Probabilistic Relational Models of On-line User Behavior, WEBKDD 2003, 2003

[3] S. Brin, L. Page: The Anatomy of a Large-Scale Hypertextual Web Search Engine, Computer Networks 30(1-7), 1998

[4] S. Brin, L. Page, R. Motwani, T. Winograd: The PageRank Citation Ranking: Bringing Order to the Web, Stanford Digital Libraries Working Paper, 1998

[5] C. Bryce, M. Pawlak, K. Tolle, P. Werner, R. Zicari: Agent-Based Services for Information Portals, 18th Annual Symposium on Applied Computing (ACM-SAC 2003), 2003

[6] S. Casteleyn, O. De Troyer, S. Brockmans: Design Time Support for Adaptive Behavior in Web Sites, 18th Annual Symposium on Applied Computing (ACM-SAC 2003), 2003

[7] T.H. Haveliwala: Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search, IEEE Transactions on Knowledge and Data Engineering, 2003

[8] B. Hays, G. Wets, K. Vanhoof: Clustering navigational patterns on a website using a Sequence Alignment Method, IJCAI’s Workshop on Intelligent Techniques for Web Personalization, Seattle, 2001

[9] B. Hays, G. Wets, K. Vanhoof: Discovering interesting navigations on a website using a Sequence Alignment Method extended with a Interestingness Measure, Workshop on Intelligent Techniques for Web Personalization, Acapulco, 2003

[10] T. Joachims, D. Freitag, T. Mitchell: WebWatcher: A Tour Guide for the World Wide Web, 1996

[11] H. Lieberman: Letizia: An Agent That Assists Web Browsing, International Joint Conference on Artificial Intelligence, 1995

[12] C.-C. Lin, M.-S. Chen: VIPAS: Virtual Link Powered Authority Search in the Web, 29th Conference on Very Large Databases (VLDB 2003), 2003

[13] A. Nanopoulos, D. Katsaros, Y. Manolopoulos: Effective Prediction of Web-user Accesses: A Data Mining Approach, WEBKDD 2001, 2001

[14] C. Shahabi, F. Banaei-Kashani, J. Faruque: A Reliable, Efficient and Scalable System for Web Usage Data Acquisition, WEBKDD 2001, 2001

[15] S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis, K. Tolle, B. Amann, I. Fundulaki, M. Scholl, A. Vercoustre: Managing RDF Metadata for Community Webs, 2nd International Workshop on the World Wide Web and Conceptual Modeling, 2000

[16] T. W. Yan, M. Jacobsen, H. Garcia-Molina, U. Dayal: From User Access Patterns to Dynamic Hypertext Linking, SIDL-WP-1996-0030, 1996